Functional Singular Spectrum Analysis: Nonparametric Decomposition and Forecasting Approaches for Functional Time Series

Jordan Christopher Trinka
Marquette University

FUNCTIONAL SINGULAR SPECTRUM ANALYSIS: NONPARAMETRIC DECOMPOSITION AND FORECASTING APPROACHES FOR FUNCTIONAL TIME SERIES

by

Jordan Trinka, B.A., M.S.

A Dissertation submitted to the faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

May 2021

ABSTRACT
FUNCTIONAL SINGULAR SPECTRUM ANALYSIS: NONPARAMETRIC DECOMPOSITION AND FORECASTING APPROACHES FOR FUNCTIONAL TIME SERIES

Jordan Trinka, B.A., M.S.

Marquette University, 2021

In this dissertation, we develop nonparametric decomposition methods and the subsequent forecasting techniques for functional, time-dependent data known as functional time series (FTS). We use ideas from functional data analysis (FDA) and singular spectrum analysis (SSA) to introduce the nonparametric decomposition method known as functional SSA (FSSA) and its associated forecasting techniques. We also extend these developed methodologies into multivariate FSSA (MFSSA) over different dimensional domains and its subsequent forecasting routines so that we may perform nonparametric decomposition and prediction of multivariate FTS (MFTS). The FSSA algorithm may be viewed as a signal extraction technique and we find that the method outperforms other competing approaches in estimating the underlying deterministic nature of an FTS. We then develop the FSSA recurrent forecasting (FSSA R-forecasting) and FSSA vector forecasting (FSSA V-forecasting) algorithms to predict future observations and we find that these methods outperform the current gold standard for nonparametric forecasting of periodic FTS. Finally, we finish with the implementation of MFSSA and the respective forecasting algorithms (MFSSA R-forecasting and MFSSA V-forecasting), which are used to decompose and forecast MFTS. We find that the MFSSA methods outperform their univariate FSSA counterparts in signal extraction and forecasting of MFTS data.

ACKNOWLEDGEMENTS

Jordan Trinka, B.A., M.S.

I would first like to acknowledge my wife Sidney. Without her love and support, none of this would have been possible and I am so grateful that I get to spend the rest of my life with the most perfect person in the world. Thank you from the bottom of my heart for always supporting me and loving me through this process.

I would like to acknowledge my family. The many sacrifices they have made has helped me have success.

I would like to acknowledge my doctoral adviser Dr. Mehdi Maadooliat. His constant encouragement and demand for high quality work helped me to become a complete scientist.

I would like to acknowledge my good friends Austin, Anmol, and Samuel. They consistently encouraged me in my work and enlightened me to new and awesome ways of thinking during our days and fun nights together.

I would like to acknowledge Dr. Hossein Haghbin. His comments and extra time he devoted helped me significantly in my development as a scientist and researcher.

I would like to acknowledge Dr. Seyed Morteza Najibi and Dr. Rahim Mahmoudvand for their contributions to the functional singular spectrum analysis project.

I would like to acknowledge Dr. Rebecca Sanders, Dr. Daniel Rowe, Dr. Gary Krenz, and Dr. Naveen Bansal for serving on my committee and always going above and beyond to give me feedback on my work and encourage me to always do better.

I would like to acknowledge Dr. Stephen Merrill who helped open up my mind to the creativity and beauty that exists in mathematics.

I would like to acknowledge the Department of Mathematical and Statistical Sciences at Marquette University. It was obvious to me that the department cared not only about my growth as a scientist but as an individual as well.

I would like to acknowledge the graduate school at Marquette University for providing me with this amazing opportunity to earn my doctoral degree from Marquette.

I would like to acknowledge the Wehr Foundation which provided funding for my dissertation research during the summers of 2019 and 2020.

I would like to acknowledge my college professors Dr. Eric Sullivan, Dr. Theodore Wendt, and Dr. Kelly Cline. They helped me realize my passion for the mathematical sciences and always encouraged me to chase my dreams.

I would like to acknowledge my 10^{th} grade Biology teacher, the late Mrs. Dawn Lawrence. She first sparked my passion for science in general and inspired me to turn my life around from an early age. She is sorely missed by all who knew her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The goal of decomposition of a dataset is to extract interesting and informative patterns that are present in data. The technique of decomposition may be used in exploratory analysis of time-independent data and the process applied to such datasets is often times referred to as dimension reduction. If the data is composed of real-valued vectors, then principal component analysis can be used to perform the dimension reduction (Hotelling, 1933; Pearson, 1901). If the observations belong to a functional vector space, then the functional principal component analysis algorithm may be used (Ramsay and Silverman, 2005). For time-dependent data, singular spectrum analysis (SSA)-based algorithms are often times employed to perform the decomposition process (Golyandina et al., 2015, 2001; Hassani and Mahmoudvand, 2013). In addition, the results of these decomposition algorithms are commonly used to perform nonparametric predictive analysis where for time-independent, real-valued data, principal component regression can be used (Massy, 1965). If the data are functional instead of real-valued, then functional principal component regression may be used (Reiss and Ogden, 2007). If data are time-dependent, prediction by SSA-based recurrent and vector forecasting approaches may be leveraged (Golyandina et al., 2001; Hassani and Mahmoudvand, 2013). While there has been extensive work done on the development and use of algorithms that perform nonparametric decomposition and prediction of the aforementioned data types, there has been a need to extend the methodologies to handle time-dependent, functional data known as a functional time series (FTS). Currently, there exists methodologies that attempt to perform the nonparametric decomposition process and subsequent forecasting of FTS data by leveraging various kinds of functional principal component analysis which assume the data are time-independent (Beyaztas and Shang, 2019; Hörmann et al., 2015; Hyndman and Ullah, 2007; Wagner-Muns et al., 2018). The goal of our present work is to develop functional extensions of the SSA algorithms that incorporate time-dependency into the decomposition process and subsequent nonparametric forecasts to do a better job in uncovering more informative patterns in data and improve predictions of stochastic processes. In order to lay the foundations of the theory that we develop, we first give a brief review of the decomposition process and subsequent prediction methods performed on time-independent non-functional and functional data as well as time-dependent non-functional data.

1.1 Dimension Reduction and Prediction of Time-Independent Data

We assume that we observe an $N \times p$ data matrix, $\mathbf{X}$, where the columns have been demeaned and each row is a multivariate observation. The goal is to leverage a singular value decomposition (SVD) of $\mathbf{X}$ to obtain a set of singular values, $\{\sqrt{\lambda_i}\}_{i=1}^r$, N-dimensional orthonormal left singular vectors, $\{\mathbf{u}_i\}_{i=1}^r$, and p-dimensional orthonormal right singular vectors, $\{\mathbf{v}_i\}_{i=1}^r$, such that

$$\mathbf{X} = \sum_{i=1}^{r} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$$

where r is the rank of $\mathbf{X}$. We also have that the bases (left and right singular vectors) explain the maximum amount of variability in the data and in order to find these basis elements, we go through the process of principal component analysis.

1.1.1 Principal Component Analysis

In the following, we describe how to find the first principal component vector, $\mathbf{v}_1 \in \mathbb{R}^p$, that explains the maximum amount of variation in the p-dimensional observations. We then describe how to find the successive principal component vectors, $\{\mathbf{v}_i\}_{i=2}^r$, that explain other modes of variation in the data.

The Leading Principal Component

The principal component analysis problem of finding $\mathbf{v}_1$ is given by

$$\underset{\|\mathbf{v}_1\|_{\mathbb{R}^p}=1}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_i^\top \mathbf{v}_1\right)^2 = \underset{\|\mathbf{v}_1\|_{\mathbb{R}^p}=1}{\operatorname{argmax}} \frac{1}{N} \mathbf{v}_1^\top \mathbf{X}^\top \mathbf{X} \mathbf{v}_1,$$

where $\|\cdot\|_{\mathbb{R}^p}$ is the Euclidean norm of $\mathbb{R}^p$ and the row vector, $\mathbf{x}_i^\top$, is the i^{th} row of $\mathbf{X}$. We find an equivalent maximization problem is given by

$$\underset{\|\mathbf{v}_1\|_{\mathbb{R}^p}=1}{\operatorname{argmax}} \mathbf{v}_1^\top \mathbf{X}^\top \mathbf{X} \mathbf{v}_1. \tag{1}$$

Since we have the constraint of $\|\mathbf{v}_1\|_{\mathbb{R}^p} = 1$, then the problem stated in equation (1) is a maximization of the Rayleigh quotient (Trefethen and Bau III, 1997) which attains a maximum value at the leading eigenvalue of the $p \times p$ matrix, $\mathbf{X}^\top \mathbf{X}$. Thus, the $\mathbf{v}_1$ that solves the problem is the leading eigenvector of $\mathbf{X}^\top \mathbf{X}$.

Successive Principal Components

For $i \geq 2$ we find successive principal component vectors, $\mathbf{v}_i$'s, by first forming the projection of the N observations onto the space orthogonal to the linear span of $\{\mathbf{v}_j\}_{j=1}^{i-1}$ and the result of the orthogonalization is given by the $N \times p$ matrix

$$\mathbf{X}_i = \mathbf{X} - \sum_{j=1}^{i-1} \mathbf{X}\mathbf{v}_j\mathbf{v}_j^\top.$$

From here, we solve the maximization problem of

$$\underset{\|\mathbf{v}_i\|_{\mathbb{R}^p}=1}{\operatorname{argmax}} \mathbf{v}_i^\top \mathbf{X}_i^\top \mathbf{X}_i \mathbf{v}_i$$

where the solution is found by setting $\mathbf{v}_i$ to be the leading eigenvector of the $p \times p$ matrix, $\mathbf{X}_i^\top \mathbf{X}_i$, and $\mathbf{v}_i$ also happens to be the i^{th} eigenvector of $\mathbf{X}^\top \mathbf{X}$.

In general, we find that the eigendecomposition of $\mathbf{X}^\top \mathbf{X}$ gives us our collection of right singular vectors, $\{\mathbf{v}_i\}_{i=1}^r$, and associated singular values, $\sqrt{\lambda_i}$, where each λ_i is the eigenvalue that corresponds with the i^{th} principal component vector. We also find that each left singular vector, $\mathbf{u}_i$, is given by $\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{X}\mathbf{v}_i$ for $i = 1, \ldots, r$. Now that we have our decomposition, we continue on with principal component regression.

1.1.2 Principal Component Regression

We assume that we observe some response vector, $\mathbf{y} \in \mathbb{R}^N$, that has been demeaned and we have the goal of using a subset of our $\mathbf{v}_i$'s to perform prediction. We select $k \leq r$ right singular vectors to construct the $p \times k$ matrix, $\mathbf{V}_k = [\mathbf{v}_1 \cdots \mathbf{v}_k]$, and we find a scalar, α, and a vector, $\boldsymbol{\beta} \in \mathbb{R}^k$, that minimizes

$$\|\mathbf{y} - \alpha\mathbf{1} - \mathbf{X}\mathbf{V}_k\boldsymbol{\beta}\|_{\mathbb{R}^N}^2$$

by using the least squares solution where $\|\cdot\|_{\mathbb{R}^N}$ is the Euclidean norm of $\mathbb{R}^N$ and $\mathbf{1}$ is the one-vector in $\mathbb{R}^N$. A popular method used to select k is cross-validation since it was found that components that explain a small amount of variation can be just as important in prediction as compared with components that explain most of the variability; see Jolliffe (1982) and references therein for more

information.

1.2 Dimension Reduction and Prediction of Functional Time-Independent Data

For dimension reduction of time-independent functional data, we begin by defining the vector space that our observations are contained in. We define the Hilbert space, $\mathbb{H} = L^2([s_1, s_p])$, and linearly independent function vectors in $\mathbb{H}$, $\{b_i\}_{i=1}^{\infty}$, that span $\mathbb{H}$. We find that each function vector, $x \in \mathbb{H}$, may be expressed as

$$x = \sum_{i=1}^{\infty} c_i b_i$$

where $\{c_i\}_{i=1}^{\infty}$ is a unique set of scalars. In computer implementation of functional data analysis (FDA) algorithms, it is common to select a subset of basis elements, $\{b_i\}_{i=1}^{d}$, and corresponding coefficients, $\{c_i\}_{i=1}^{d}$, to approximate x where d is a positive integer. A similar type of approximation of functional observations is leveraged in the following.

We assume that we observe an $N \times p$ matrix, $\mathbf{X}$, where each column has a mean of zero and each row vector, $\mathbf{x}_i^\top$, of $\mathbf{X}$ is the i^{th} functional observation vector, $x_i \in \mathbb{H}$, sampled over the regular grid, $\{s_1, \ldots, s_p\}$, for $i = 1, \ldots, N$. We let $\mathbf{B}$ be a $p \times d$ B-spline basis matrix where $\mathbf{B}_{i,j} = \mathbf{b}_j(s_i)$ is the j^{th} basis element evaluated at a point $s_i \in \{s_1, \ldots, s_p\}$. We calculate the $N \times d$ matrix, $\mathbf{XB}$, where the i^{th} row contains the coefficients used to approximate x_i. Our goal is to find an SVD of $\mathbf{XB}$ to obtain the expression

$$\mathbf{XB} = \sum_{i=1}^{r} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$$

where $\{\sqrt{\lambda_i}\}_{i=1}^{r}$ is the collection of singular values, $\{\mathbf{u}_i\}_{i=1}^{r}$ is a collection of orthonormal left singular vectors in $\mathbb{R}^N$, and $\{\mathbf{v}_i\}_{i=1}^{r}$ is a collection of orthonormal right singular vectors in $\mathbb{R}^d$. As according to Reiss and Ogden (2007), each $\mathbf{v}_i$ contains the coefficients used to approximate the i^{th} functional principal component vector, $v_i \in \mathbb{H}$. We also find that each $\mathbf{u}_i$ captures covariance information between each of the N functional observation vectors and v_i. We may use the functional principal component method to obtain each $\mathbf{v}_i$.

1.2.1 Functional Principal Component Analysis

We first describe how to find the vector of coefficients, $\mathbf{v}_1$, that are used to approximate v_1, where v_1 explains the maximum amount of variation in the functional observation vectors (x_i's). We then

describe how to find successive vectors of coefficients, $\{\mathbf{v}_i\}_{i=2}^r$.

The Leading Functional Principal Component

Based on ideas expressed in principal component anlaysis, we formulate the problem of finding $\mathbf{v}_1$ as

$$\underset{\|v_1\|_{\mathbb{H}}=1}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_i^\top \mathbf{B}\mathbf{v}_1\right)^2 = \underset{\|v_1\|_{\mathbb{H}}=1}{\operatorname{argmax}} \frac{1}{N} \mathbf{v}_1^\top \mathbf{B}^\top \mathbf{X}^\top \mathbf{X}\mathbf{B}\mathbf{v}_1$$

where $\|\cdot\|_{\mathbb{H}}$ is the norm induced by the inner product equipped to $\mathbb{H}$. From here, we can simplify the maximization problem to

$$\underset{\|v_1\|_{\mathbb{H}}=1}{\operatorname{argmax}} \mathbf{v}_1^\top \mathbf{B}^\top \mathbf{X}^\top \mathbf{X}\mathbf{B}\mathbf{v}_1. \tag{2}$$

Similar to principal component analysis, the problem shown in equation (2) is solved by setting $\mathbf{v}_1$ to be the leading eigenvector of the $d \times d$ matrix, $\mathbf{B}^\top \mathbf{X}^\top \mathbf{X}\mathbf{B}$.

Successive Functional Principal Components

For $i \geq 2$ we find successive vectors, $\mathbf{v}_i$'s, by estimating the projection of the functional observation vectors (x_i's) onto the space orthogonal to the linear span of the functional principal component vectors, $\{v_j\}_{j=1}^{i-1}$. The coefficients of the estimates that stem from the aforementioned orthogonalization process are given in the rows of the following $N \times d$ matrix,

$$(\mathbf{X}\mathbf{B})_i = \mathbf{X}\mathbf{B} - \sum_{j=1}^{i-1} \mathbf{X}\mathbf{B}\mathbf{v}_j\mathbf{v}_j^\top.$$

From here, we solve the maximization problem of

$$\underset{\|v_i\|_{\mathbb{H}}=1}{\operatorname{argmax}} \mathbf{v}_i^\top \left(\mathbf{X}\mathbf{B}\right)_i^\top \left(\mathbf{X}\mathbf{B}\right)_i \mathbf{v}_i$$

where the solution is the leading eigenvector of the $d \times d$ matrix, $\left(\mathbf{X}\mathbf{B}\right)_i^\top \left(\mathbf{X}\mathbf{B}\right)_i$, which also happens to be the i^{th} eigenvector of $\mathbf{B}^\top \mathbf{X}^\top \mathbf{X}\mathbf{B}$.

We see from this discussion that the eigendecomposition of $\mathbf{B}^\top \mathbf{X}^\top \mathbf{X}\mathbf{B}$ gives us the right singular vectors, $\{\mathbf{v}_i\}_{i=1}^r$, and associated singular values, $\{\sqrt{\lambda_i}\}_{i=1}^r$, where each λ_i is the eigenvalue that corresponds with the i^{th} functional principal component vector. We also find that the left singular

vectors are given by $\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{XBv}_i$ for $i = 1, \ldots, r$. Now that we have our decomposition, we continue with functional principal component regression.

1.2.2 Functional Principal Component Regression

For some random response vector, $\mathbf{y} \in \mathbb{R}^N$, that has been demeaned, we have the goal of using a subset of our $\mathbf{v}_i$'s to predict future observations. Similar to principal component regression, we select $k \leq r$, we form the $d \times k$ matrix, $\mathbf{V}_k = [\mathbf{v}_1 \cdots \mathbf{v}_k]$, and we find a scalar, α, and vector, $\boldsymbol{\beta} \in \mathbb{R}^k$, such that

$$\|\mathbf{y} - \alpha\mathbf{1} - \mathbf{XBV}_k\boldsymbol{\beta}\|^2_{\mathbb{R}^N}$$

is minimized by using the least squares solution where $\|\cdot\|_{\mathbb{R}^N}$ is the Euclidean norm of $\mathbb{R}^N$ and $\mathbf{1}$ is the one-vector in $\mathbb{R}^N$. Similar to principal component regression, we select k by leveraging cross-validation. From here, we discuss decomposition of time-dependent data via SSA and the resulting nonparametric forecasting methods.

1.3 Nonparametric Decomposition and Forecasting of Time-Dependent Data

For nonparametric decomposition of univariate time series, we may utilize SSA (Golyandina et al., 2001) and in addition, we also have that the results of this methodology can then be used to perform nonparametric forecasting of time-dependent data. To perform nonparametric decomposition and forecasting of multivariate time series we leverage multivariate SSA (MSSA) which has been developed in two different approaches named vertical MSSA (VMSSA) and horizontal MSSA (HMSSA), see Hassani and Mahmoudvand (2013) and Golyandina et al. (2015) for more information. For now, we continue with the univariate SSA methodology and subsequent forecasting algorithms.

1.3.1 Univariate Singular Spectrum Analysis

One of the popular approaches in the decomposition of time series is accomplished using rates of change. In this approach, the observed time series is partitioned (decomposed) into informative trends plus potential seasonal (cyclical) and noise (irregular) components. Aligned with this principle, SSA is a model-free procedure that is commonly used as a nonparametric technique in analyzing time series. SSA is intrinsically motivated as an exploratory and model building tool rather than a confirmatory procedure (Golyandina et al., 2001). SSA does not require restrictive assumptions

such as stationarity, linearity, and normality. It can be used for a wide range of purposes such as trend and periodic component detection and extraction, smoothing, forecasting, change-point detection, gap filling, causality and so on; (see, e.g. Golyandina et al., 2001; Golyandina and Osipov, 2007; Kondrashov et al., 2010; Mahmoudvand and Rodrigues, 2016; Mohammad and Nishida, 2011; Moskvina and Zhigljavsky, 2003; Rodrigues and Mahmoudvand, 2016).

The goal of SSA is to extract out modes of variation in time series data. It provides a representation of the given time series in terms of rank one approximations generated from an SVD of a so-called trajectory matrix (Alexandrov, 2009). These rank one approximations are built from so-called eigentriples where an eigentriple is defined in SSA literature as a set that contains a singular value, corresponding left singular vector, and corresponding right singular vector that are found using the SVD technique (Golyandina et al., 2001; Hassani and Mahmoudvand, 2013). Presently, many studies have been published with extensions and applications of SSA. The extension to MSSA is straightforward (Golyandina et al., 2015, 2018; Golyandina and Zhigljavsky, 2013; Hassani and Mahmoudvand, 2013, 2018), and an extension of SSA to a two-dimensional setting can be found in Golyandina et al. (2015) and references therein. For now, we continue on with our review of the SSA algorithm.

Throughout this section, we consider y_i's as elements of the Euclidean space, $\mathbb{R}$. Suppose that $\mathbf{y}_N = [y_1, y_2, \ldots, y_N]^\top$ is a realization of size N from a time series. The basic SSA algorithm consists of four steps: Embedding, Decomposition, Grouping, and Reconstruction. The purpose of the first step of embedding is to generate a trajectory matrix that corresponds with the time series where the matrix can be decomposed using an SVD.

Step 1. Embedding

This step generates a matrix by tracking a moving window of size L over the original time series, where L is an integer called the window length parameter and $1 < L < N/2$. Embedding can be regarded as a mapping operator, $\mathcal{T}$, that transfers the series, $\mathbf{y}_N$, into a so-called trajectory matrix, $\mathbf{X}$, of dimension $L \times K$, defined by

$$\mathbf{X} = \mathcal{T}(\mathbf{y}_N) = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_K],$$

where $K = N - L + 1$ and $\boldsymbol{x}_j = [y_j, y_{j+1}, \ldots, y_{j+L-1}]^\top \in \mathbb{R}^L$ is known as the j^{th}, L-lagged vector for $j = 1, \ldots, K$. Note that the trajectory matrix, $\mathbf{X}$, is a Hankel matrix, which means that the

elements along the anti-diagonals are equal. Indeed, the embedding operator, $\mathcal{T}$, is a one-to-one mapping from $\mathbb{R}^N$ into $\mathbb{R}_H^{L\times K}$ where $\mathbb{R}_H^{L\times K}$ is the space of all $L \times K$ Hankel matrices in the space of $L \times K$ real-valued matrices. In addition, we also have that the columns of $\mathbf{X}$ describe time series behavior over sub-intervals of time and we now extract out the dominant modes of variation shared between these L-lagged vectors in the following step.

Step 2. Decomposition

In this step, an SVD for the trajectory matrix, $\mathbf{X}$, is computed as:

$$\mathbf{X} = \sum_{i=1}^{r} \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^\top = \sum_{i=1}^{r} \mathbf{X}_i, \tag{3}$$

where r is the rank of $\mathbf{X}$, $\sqrt{\lambda_i}$ is the i^{th} singular value, $\mathbf{u}_i \in \mathbb{R}^L$ is the associated left singular vector, and $\mathbf{v}_i \in \mathbb{R}^K$ is the associated right singular vector of $\mathbf{X}$. In addition, we define the set, $\{\sqrt{\lambda_i}, \mathbf{u}_i, \mathbf{v}_i\}$, as the i^{th} eigentriple of $\mathbf{X}$, and we use the elements of this eigentriple to construct the i^{th} rank one elementary matrix of dimension $L \times K$ given by $\mathbf{X}_i = \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^\top$. In the next step, we form disjoint groups of eigentriples where the eigentriples in each group describe similar time series behavior. We then add rank one elementary matrices together within each group where each elementary matrix is built using the corresponding eigentriple.

Step 3. Grouping

Consider a partition of the set of indices, $\{1, 2, \ldots, r\}$, into m disjoint subsets, $\{I_1, I_2, \ldots, I_m\}$. For any positive integer q, i.e. $1 \leq q \leq m$, the $L \times K$ matrix, $\mathbf{X}_{I_q}$, is defined as $\mathbf{X}_{I_q} = \sum_{i\in I_q} \mathbf{X}_i$. Thus, by the expansion of equation (3), we have the grouped matrix decomposition

$$\mathbf{X} = \mathbf{X}_{I_1} + \mathbf{X}_{I_2} + \cdots + \mathbf{X}_{I_m}. \tag{4}$$

Each group in equation (4) should correspond to a component in the time series decomposition such as trend, seasonal, noise, etc. Exploratory plots of eigentriple elements (such as plots of singular values, left/right singular vectors, etc.) can be leveraged as helpful tools when making decisions on how to form the disjoint groups (Golyandina et al., 2001). In the final step of SSA, we extract out a time series from each $\mathbf{X}_{I_q}$.

Step 4. Reconstruction

The resulting matrices, $\mathbf{X}_{I_q}$, are transformed back into the form of the original series $\mathbf{y}_N$ by the inverse operator $\mathcal{T}^{-1} : \mathbb{R}_H^{L\times K} \to \mathbb{R}^N$. In order to do this, it is necessary that each $\mathbf{X}_{I_q}$ is approximated by a matrix in $\mathbb{R}_H^{L\times K}$. This approximation is performed optimally in the sense of orthogonal projection of $\mathbf{X}_{I_q}$ onto $\mathbb{R}_H^{L\times K}$ with respect to the Frobenius norm. Denote this projection by $\Pi_H : \mathbb{R}^{L\times K} \to \mathbb{R}_H^{L\times K}$. It is shown that applying Π_H to some $L \times K$ matrix is the same as averaging of the matrix elements over the antidiagonals (where $i + j =$ constant). As such, we may define the $L \times K$ matrix, $\widetilde{\mathbf{X}}_{I_q} = \Pi_H \mathbf{X}_{I_q}$, as the projection of $\mathbf{X}_{I_q}$ onto $\mathbb{R}_H^{L\times K}$ and from here, we define the q^{th} reconstructed time series of length N as $\tilde{\mathbf{y}}_N^q = \mathcal{T}^{-1}\left(\widetilde{\mathbf{X}}_{I_q}\right)$ for $q = 1, \ldots, m$. We have that each $\mathbf{y}_N^q$ describes a mode of variation present in the original series, $\mathbf{y}_N$ such as mean, periodicity, etc.

It is well known that SSA does not require restrictive assumptions; however, it is ideal to have a time series with additive components. If a time series is composed of such additive components, then we call the series separable. Therefore, we present tools to measure the separability of components in the next subsection.

Separability

Let $\mathbf{y}_N = [y_1, \ldots, y_N]^\top$ and $\mathbf{z}_N = [z_1, \ldots, z_N]^\top$ be two time series. The weighted-correlation (w-correlation) between $\mathbf{y}_N$ and $\mathbf{z}_N$ is defined as

$$\rho^{(w)}(\mathbf{y}_N, \mathbf{z}_N) = \frac{\langle \mathbf{y}_N, \mathbf{z}_N \rangle_w}{\|\mathbf{y}_N\|_w \|\mathbf{z}_N\|_w} \tag{5}$$

where $\langle \mathbf{y}_N, \mathbf{z}_N \rangle_w = \sum_{i=1}^N w_i y_i z_i$, $w_i = \min\{i, L, N-i+1\}$, and $\|\mathbf{y}_N\|_w = \sqrt{\langle \mathbf{y}_N, \mathbf{y}_N \rangle_w}$. We call $\mathbf{y}_N$ and $\mathbf{z}_N$ w-orthogonal if $\rho^{(w)}(\mathbf{y}_N, \mathbf{z}_N) = 0$ for appropriate values of L (see the next subsection for more details). Note that $\tilde{\mathbf{y}}_N^q$, $q = 1, \ldots, r$, is the reconstructed component produced by the group, I_q, and the matrix, $\boldsymbol{\rho}^{(w)} = \left[\rho^{(w)}(\tilde{\mathbf{y}}_N^i, \tilde{\mathbf{y}}_N^j)\right]_{i,j=1}^r$, is called the w-correlation matrix (Golyandina et al., 2001). If $\tilde{\mathbf{y}}_N^i$ and $\tilde{\mathbf{y}}_N^j$ are approximately w-orthogonal, then it is recommended that we perform the grouping stage of SSA in such a way where i and j belong to different groups of indices so that we do not mix together dissimilar modes of variation in our reconstructions. It is clear that the w-correlation matrix can help in determining how to perform the grouping stage of SSA. The determination of L and the grouping parameters is essential in the SSA algorithm and we now discuss techniques used to select values for these parameters.

Parameter Selection

There are two basic parameters in the SSA procedure; window length (L) and grouping parameters. Choosing improper values for these parameters yields an incomplete reconstruction and misleading results in subsequent analysis. In spite of the importance of choosing L and grouping parameters for SSA, no ideal solution has been proposed yet. A thorough review of the problem shows that the optimal choice of the parameters depends on the intrinsic structure of the data and the purposes of the study (Golyandina et al., 2001; Golyandina and Zhigljavsky, 2013). However, there are several recommendations and rules that work well for a wide range of scenarios. It is recommended to select the window length parameter, L, to be a large integer that is multiple of the periodicities of the time series, but not larger than $N/2$.

In addition, there are several methods for effective grouping. These techniques include analyzing the periodogram, paired plot of the singular vectors, scree plot of the singular values, and w-correlation plot; see Golyandina et al. (2001) for more details. We also note that these techniques used in parameter selection for the SSA routine hold true for all forms of the algorithm including the multivariate, functional, and multivariate functional versions to be discussed later. From here, we continue on with SSA forecasting of time series data.

1.3.2 Univariate Singular Spectrum Analysis Forecasting

For the SSA forecasting methods, known as SSA recurrent forecasting and SSA vector forecasting, we are primarily concerned with making predictions of the estimate of the true, underlying signal that we extract from SSA. To this end, we perform the grouping and reconstruction stages in a way such that we may extract two time series, the first that captures the deterministic signal that is assumed to be present in $\mathbf{y}_N$ and the second that captures the residual.

SSA Grouping for Forecasting

Consider $\mathfrak{S}$ to be a subset of indices, $\{1, \ldots, r\}$, and define the $L \times K$ matrix, $\mathbf{X}_{\mathfrak{s}} = \sum_{i \in \mathfrak{S}} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$. We assume that $\mathfrak{S}$ represents a decomposition of the signal. This allows us to write

$$\mathbf{X} = \mathbf{X}_{\mathfrak{s}} + \mathbf{X}_{\mathfrak{n}},$$

where $\mathbf{X}_{\mathfrak{n}} = \sum_{i \in \mathfrak{S}^c} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$ is an $L \times K$ matrix representative of the residual component. We also denote the j^{th} length L column vector of $\mathbf{X}_{\mathfrak{s}}$ with $\mathbf{x}_j^{\mathfrak{s}} \in \mathbb{R}^L$ which is leveraged in SSA vector

forecasting. In the reconstruction step, we extract a time series from $\mathbf{X}_{\mathfrak{s}}$ to be used in SSA recurrent forecasting.

SSA Reconstruction for Forecasting

In this step we use $\mathcal{T}^{-1} : \mathbb{R}_H^{L\times K} \to \mathbb{R}^N$ to transform back $\mathbf{X}_{\mathfrak{s}}$ to a time series of length N, $\tilde{\mathbf{y}}_N = [\tilde{y}_1, \ldots, \tilde{y}_N]^\top \in \mathbb{R}^N$, that captures the deterministic nature of $\mathbf{y}_N$. Just like in the reconstruction stage of Subsection 1.3.1, we Hankelize $\mathbf{X}_{\mathfrak{s}}$ by performing antidiagonal averaging of the elements to form the $L \times K$ matrix, $\widetilde{\mathbf{X}} = \Pi_H \mathbf{X}_{\mathfrak{s}}$, and from here we find that $\tilde{\mathbf{y}}_N = \mathcal{T}^{-1}\widetilde{\mathbf{X}}$. Now that we have established the grouping and reconstruction stages of SSA with the goal of forecasting, we introduce some preliminary notations to be used in the algorithms.

SSA Forecasting Preliminaries

We denote $\mathbf{x}^\nabla \in \mathbb{R}^{L-1}$ and $\mathbf{x}^\Delta \in \mathbb{R}^{L-1}$ to be the vectors that contain the first and last (respectively) $L-1$ components of some vector $\mathbf{x} \in \mathbb{R}^L$. For a positive integer $k < r$, we define the vector spaces, $\mathbb{L} = \text{span}\{\mathbf{u}_i\}_{i=1}^k$ and $\mathbb{L}^\nabla = \text{span}\{\mathbf{u}_i^\nabla\}_{i=1}^k$, and we define the scalar, $\pi_i \in \mathbb{R}$, to be the last component from the i^{th} left singular vector, $\mathbf{u}_i$. Next we define the scalar, $\nu^2 = \sum_{i=1}^k \pi_i^2$, and $\mathbf{g}_{N+M} = [g_1, \ldots, g_N, g_{N+1}, \ldots, g_{N+M}]^\top \in \mathbb{R}^{N+M}$ to be a time series of length $N+M$ where the first N terms are close to $\{\tilde{y}_i\}_{i=1}^N$ and the remaining M terms are determined in the following forecasting routines.

SSA Recurrent Forecasting

The idea behind the SSA recurrent forecasting algorithm is to express the prediction as a linear combination of the previous $L-1$ reconstructed elements. As according to Golyandina et al. (2001), we first define the vector $\mathbf{e} = [0, \ldots, 0, 1]^\top \in \mathbb{R}^L$ and we find that as long as $\mathbf{e} \notin \mathbb{L}$, then we obtain the following algorithm.

Algorithm A - SSA Recurrent Forecasting Algorithm:

$$g_i = \begin{cases} \tilde{y}_i, & i = 1, \ldots, N \\ \sum_{j=1}^{L-1} a_j g_{i+j-L}, & i = N+1, \ldots, N+M \end{cases} \tag{6}$$

where the coefficients, $\{a_j\}_{j=1}^{L-1}$, are given in the $(L-1)$-dimensional vector,
$\mathcal{R} = [a_1, \ldots, a_{L-1}]^\top = \frac{1}{1-\nu^2}\sum_{n=1}^k \pi_n \mathbf{u}_n^\nabla$. Clearly from the definition of $\mathcal{R}$, we see that ν^2 must be

less than one and this is satisfied by the assumption that $\mathbf{e} \notin \mathbb{L}$. For brevity, we call this method the *SSA R-forecasting* algorithm.

SSA Vector Forecasting

The recurrent method has been extended to the vector forecasting realm allowing us to predict observations using L-dimensional vectors. Our goal is to find a linear operator $\mathbf{V} : \mathbb{L} \to \mathbb{L}$ such that for some L-dimensional vector, $\mathbf{y}$, defined by $\mathbf{y} = \mathbf{V}\mathbf{x}$, we have that the distance between the $(L-1)$-dimensional vectors, $\mathbf{y}^{\nabla}$ and $\mathbf{x}^{\Delta}$, is minimal where $\mathbf{x} = [x_1, \ldots, x_L]^{\top}$ is an L-dimensional vector in $\mathbb{L}$. This minimization is performed optimally in the sense of an orthogonal projection onto $\mathbb{L}^{\nabla}$. To this end, we define the $(L-1) \times k$ matrix, $\mathbf{U}^{\nabla} = \left[\mathbf{u}_1^{\nabla} \cdots \mathbf{u}_k^{\nabla}\right]$, and now we define the $(L-1) \times (L-1)$ least squares orthogonal projection matrix, that projects $L-1$ vectors onto $\mathbb{L}^{\nabla}$, as

$$\Pi = \mathbf{U}^{\nabla} \left(\left(\mathbf{U}^{\nabla}\right)^{\top} \mathbf{U}^{\nabla} \right)^{-1} \left(\mathbf{U}^{\nabla}\right)^{\top}.$$

We immediately obtain the first $L-1$ components of vector $\mathbf{y}$ as $\mathbf{y}^{\nabla} = \Pi \mathbf{x}^{\Delta}$. In order to obtain the last component, y_L, of $\mathbf{y} = [y_1, \ldots, y_L]^{\top}$, we leverage recurrent forecasting by assuming that $\mathbf{e} \notin \mathbb{L}$ and we obtain

$$y_L = \sum_{j=1}^{L-1} a_j y_j, \quad \mathcal{R} = [a_1, \ldots, a_{L-1}]^{\top} = \frac{1}{1-\nu^2} \sum_{n=1}^{k} \pi_n \mathbf{u}_i^{\nabla}.$$

It is also shown in Golyandina et al. (2001) that $y_L = \sum_{j=1}^{L-1} a_j x_{j+1}$. Now we assume that $\mathbf{e} \notin \mathbb{L}$ and we find that

$$\mathbf{y} = \mathbf{V}\mathbf{x} = \begin{bmatrix} \Pi \mathbf{x}^{\Delta} \\ \mathcal{R}^{\top} \mathbf{x}^{\Delta} \end{bmatrix}.$$

From here, we apply $\mathbf{V}$ to the columns of the matrix, $\mathbf{X}_{\mathfrak{s}}$, in order to predict future length L column vectors which are leveraged in the following forecasting algorithm.

Algorithm B - SSA Vector Forecasting Algorithm:

1. Define the length L column vectors

$$\mathbf{w}_j = \begin{cases} \mathbf{x}_j^{\mathfrak{s}}, & j = 1, \ldots, K \\ \mathbf{V}\mathbf{w}_{j-1}, & j = K+1, \ldots, K+M \end{cases}. \tag{7}$$

2. Form the $L \times (K+M)$ matrix $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{K+M}]$.

3. Hankelize $\mathbf{W}$ by applying Π_H in order to extract the time series $\mathbf{g}_{N+M}$.

4. The terms, $\{g_{N+1}, \ldots, g_{N+M}\}$, form the SSA vector forecast prediction.

For brevity, we call this method the *SSA V-forecasting* algorithm. Now that we have reviewed the full SSA routine, SSA R-forecasting, and SSA V-forecasting, we move into a review of the MSSA method.

1.3.3 Multivariate Singular Spectrum Analysis

Given p univariate time series of length N, $\{y_i^{(j)}\}_{i=1,\ldots,N}^{j=1,\ldots,p}$, a multivariate time series can be considered as a series of length N of p-dimensional vectors, $\vec{y}_i = \left[y_i^{(1)}, \ldots, y_i^{(p)}\right]^\top$. This allows us to write the multivariate time series in the form of the $N \times p$ matrix, $\mathbf{y}_N = [\vec{y}_1 \cdots \vec{y}_N]^\top \in \mathbb{R}^{N \times p}$. In the first step of MSSA, we form a matrix that we may apply an SVD to with the goal of extracting out time-dependent modes of variation.

Step 1. Embedding

We choose an integer L, where $1 < L < N/2$, set $K = N - L + 1$, and create the set of $L \times K$, univariate trajectory matrices, $\{\mathbf{X}^{(j)}\}_{j=1}^p$. These trajectory matrices have the form

$$\mathbf{X}^{(j)} = \left[\boldsymbol{x}_1^{(j)} \cdots \boldsymbol{x}_K^{(j)}\right], \tag{8}$$

where $\boldsymbol{x}_k^{(j)} = \left[y_k^{(j)}, \ldots, y_{k+L-1}^{(j)}\right]^\top \in \mathbb{R}^L$ is referred to as the k^{th} L-lagged vector of $\mathbf{y}_N$, associated with variable j. In HMSSA, we concatenate the univariate trajectory matrices horizontally to obtain an $L \times pK$ multivariate trajectory matrix

$$\mathbf{X} = \left[\mathbf{X}^{(1)} \cdots \mathbf{X}^{(p)}\right], \tag{9}$$

where as in the VMSSA, we concatenate those univariate trajectory matrices vertically to obtain the associated $pL \times K$ multivariate trajectory matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(1)} \\ \vdots \\ \mathbf{X}^{(p)} \end{bmatrix}. \tag{10}$$

Recall from the embedding step of Subsection 1.3.1 that a Hankel matrix is a matrix whose antidiagonal elements are equivalent. Since each univariate trajectory matrix, $\mathbf{X}^{(j)}$, is Hankel, the multivariate trajectory matrices generated by the embedding step of HMSSA and VMSSA are block Hankel.

As shown in Subsection 4.4.2, there is an interchangeable relationship between the functional extension of VMSSA and the multivariate functional extension to SSA. Without loss of generality, in the remainder of this section we review VMSSA and so hereafter, we define the $pL \times K$ multivariate trajectory matrix, $\mathbf{X} : \mathbb{R}^K \to \mathbb{R}^{pL}$, to be that of equation (10). Often times, this embedding step is viewed as applying an invertible transformation $\mathcal{T} : \mathbb{R}^{N \times p} \to \mathbb{R}^{pL \times K}$ such that

$$\mathbf{X} = \mathcal{T}(\mathbf{y}_N).$$

In the next step, we extract out time-dependent modes of variation from $\mathbf{X}$.

Step 2. Decomposition

We obtain an SVD of the rank $r \in \mathbb{N}$ multivariate trajectory matrix, $\mathbf{X}$, given by

$$\mathbf{X} = \sum_{i=1}^{r} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i=1}^{r} \mathbf{X}_i,$$

where $\{\sqrt{\lambda_i}\}_{i=1}^r$ are the singular values of $\mathbf{X}$, $\{\mathbf{u}_i\}_{i=1}^r$ are the orthonormal left singular vectors in $\mathbb{R}^{pL}$, and $\{\mathbf{v}_i\}_{i=1}^r$ are the orthonormal right singular vectors in $\mathbb{R}^K$. We define the eigentriples in the same manner as in univariate SSA and from each eigentriple, we calculate a rank one $pL \times K$ elementary matrix, $\mathbf{X}_i$, to be used in grouping. We also note that each left singular vector is $p(L-1)$-dimensional and has the form

$$\mathbf{u}_i = \left[u_{1,i}^{(1)}, \ldots, u_{L,i}^{(1)} u_{1,i}^{(2)}, \ldots, u_{L,i}^{(2)}, \ldots, u_{1,i}^{(p)}, \ldots, u_{L,i}^{(p)} \right]^\top$$

and this definition of each $\mathbf{u}_i$ is leveraged in VMSSA forecasting. In step three, we group eigentriples together that describe similar multivariate time series behavior such as mean, periodicity, trend, etc.

Step 3. Grouping

For grouping we partition the set of indices, $\{1, 2, \ldots, r\}$, into m disjoint subsets, $\{I_1, I_2, \ldots, I_m\}$, such that for any positive integer $q = 1, \ldots, m$, the $pL \times K$ matrix, $\mathbf{X}_{I_q}$, is defined as $\mathbf{X}_{I_q} = \sum_{i \in I_q} \mathbf{X}_i$. This allows us to write the original multivariate trajectory matrix, $\mathbf{X}$, as

$$\mathbf{X} = \mathbf{X}_{I_1} + \mathbf{X}_{I_2} + \cdots + \mathbf{X}_{I_m}. \tag{11}$$

Just like in the univariate case, the grouping should be done so that each $\mathbf{X}_{I_q}$ describes a different feature of the original multivariate time series such as trend or seasonality. The grouping can be achieved by looking at exploratory plots like paired-plots or scree plots (Golyandina et al., 2001; Hassani and Mahmoudvand, 2018). In step four, we extract out a multivariate time series from each $\mathbf{X}_{I_q}$ that describes a different component of variation present in $\mathbf{y}_N$.

Step 4. Reconstruction

Note that the $\mathbf{X}_{I_q}$'s ($q = 1, \cdots, m$), given in equation (11), are not necessarily block Hankel, and therefore we cannot use the $\mathcal{T}^{-1}$ transformation to extract out a multivariate time series from each $\mathbf{X}_{I_q}$. A popular remedy in the literature is to use the orthogonal projection approach and approximate $\mathbf{X}_{I_q}$'s with appropriate block Hankel matrices.

The matrix $\mathbf{X}_{I_q}$ can be written in the block form:

$$\mathbf{X}_{I_q} = \begin{bmatrix} \mathbf{X}_{I_q}^{(1)} \\ \vdots \\ \mathbf{X}_{I_q}^{(p)} \end{bmatrix},$$

where $\mathbf{X}_{I_q}^{(j)}$ is an $L \times K$ matrix for $j = 1, \ldots, p$. The orthogonal projection of $\mathbf{X}_{I_q}$ onto the space of the block Hankel matrices is done by averaging the antidiagonal elements of each $\mathbf{X}_{I_q}^{(j)}$ just like in the reconstruction stage of Subsection 1.3.1. We denote this approximated $L \times K$ block Hankel matrix as $\widetilde{\mathbf{X}}_{I_q}$, and use the inverse transformation, $\mathcal{T}^{-1} : \mathbb{R}_H^{pL \times K} \to \mathbb{R}^{N \times p}$, to obtain the q^{th} reconstructed multivariate time series

$$\tilde{\mathbf{y}}_N^q = \mathcal{T}^{-1}(\widetilde{\mathbf{X}}_{I_q}) \in \mathbb{R}^{N \times p}.$$

As in univariate SSA, we develop a measure of separability for reconstructed multivariate time series.

Separability

Let $\mathbf{y}_N$ and $\mathbf{z}_N$ be two multivariate time series of length N. The w-correlation between $\mathbf{y}_N$ and $\mathbf{z}_N$ is defined in a similar manner as seen in equation (5) only now we define $\langle \mathbf{y}_N, \mathbf{z}_N \rangle_w = \sum_{j=1}^{p} \sum_{i=1}^{N} w_i y_i^{(j)} z_i^{(j)}$. The definition of the scalar, w_i, follows directly from the univariate SSA discussion on separability. In addition the definition of the norm, $\|\mathbf{y}_N\|_w$, the discussion on w-orthogonality of $\mathbf{y}_N$ and $\mathbf{z}_N$, and the definition of the w-correlation matrix is straightforward from the univariate SSA discussion on separability.

1.3.4 Vertical Multivariate Singular Spectrum Analysis Forecasting

In the VMSSA forecasting algorithms, called VMSSA recurrent forecasting and VMSSA vector forecasting, we are primarily concerned with making predictions of the estimate of the true deterministic signal that is assumed to be in $\mathbf{y}_N$. We extract the signal using the VMSSA method and in the following we develop the grouping and reconstruction stages with this goal in mind.

VMSSA Grouping for Forecasting

Consider $\mathfrak{S}$ to be a subset of indices, $\{1, \ldots, r\}$, and define the $pL \times K$ matrix, $\mathbf{X}_{\mathfrak{s}} = \sum_{i \in \mathfrak{S}} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$, where $\mathfrak{S}$ is a decomposition of the signal. This allows us to write

$$\mathbf{X} = \mathbf{X}_{\mathfrak{s}} + \mathbf{X}_{\mathfrak{n}},$$

where the $pL \times K$ matrix, $\mathbf{X}_{\mathfrak{n}} = \sum_{i \in \mathfrak{S}^c} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^\top$, corresponds with the residual component. We also denote the j^{th}, pL-dimensional column vector of $\mathbf{X}_{\mathfrak{s}}$ with $\mathbf{x}_j^{\mathfrak{s}}$ which is leveraged in the VMSSA vector forecasting algorithm. In the next step, we extract a multivariate time series from $\mathbf{X}_{\mathfrak{s}}$ that is used in the VMSSA recurrent forecasting method.

VMSSA Reconstruction for Forecasting

Our goal is to use $\mathcal{T}^{-1} : \mathbb{R}^{pL \times K} \to \mathbb{R}^{N \times p}$ to transform back $\mathbf{X}_{\mathfrak{s}}$ to a multivariate time series, $\tilde{\mathbf{y}}_N = \left[\vec{\tilde{y}}_1 \cdots \vec{\tilde{y}}_N\right]^\top \in \mathbb{R}^{N \times p}$, that captures the deterministic nature of $\mathbf{y}_N$ where $\vec{\tilde{y}}_i$ is a p-dimensional column vector. We follow the block Hankelization routine decribed in the reconstruction stage of Subsection 1.3.3 and we denote the orthogonal projection of $\mathbf{X}_{\mathfrak{s}}$ onto the space of $pL \times K$ block Hankel matrices as $\widetilde{\mathbf{X}}$. From here, we find that $\tilde{\mathbf{y}}_N = \mathcal{T}^{-1}\widetilde{\mathbf{X}}$. Now that we have established the grouping

and reconstruction stages of VMSSA with the goal of forecasting, we introduce some preliminary notations to be used in the algorithms.

VMSSA Forecasting Preliminaries

We define

$$\mathbf{x}^{\nabla} = \left[x_1^{(1)}, \ldots, x_{L-1}^{(1)} x_1^{(2)}, \ldots, x_{L-1}^{(2)}, \ldots, x_1^{(p)}, \ldots, x_{L-1}^{(p)}\right]^{\top}$$
$$\mathbf{x}^{\Delta} = \left[x_2^{(1)}, \ldots, x_L^{(1)} x_2^{(2)}, \ldots, x_L^{(2)}, \ldots, x_2^{(p)}, \ldots, x_L^{(p)}\right]^{\top}$$

to be the vectors in $\mathbb{R}^{p(L-1)}$ that contain the first and last (respectively) $L-1$ components for each variable of vector $\mathbf{x} = \left[x_1^{(1)}, \ldots, x_L^{(1)} x_1^{(2)}, \ldots, x_L^{(2)}, \ldots, x_1^{(p)}, \ldots, x_L^{(p)}\right]^{\top} \in \mathbb{R}^{pL}$. For a positive integer $k < r$, we define the vector spaces $\mathbb{L} = \text{span}\{\mathbf{u}_i\}_{i=1}^{k}$ and $\mathbb{L}^{\nabla} = \text{span}\{\mathbf{u}_i^{\nabla}\}_{i=1}^{k}$, and we define the p-dimensional vector, $\vec{u}_{j,n} = \left[u_{j,n}^{(1)}, \ldots, u_{j,n}^{(p)}\right]^{\top}$. Now we define the $p \times k$ matrix

$$\mathbf{Z} = \begin{bmatrix} \pi_1^{(1)} & \pi_2^{(1)} & \cdots & \pi_k^{(1)} \\ \pi_1^{(2)} & \pi_2^{(2)} & \cdots & \pi_k^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ \pi_1^{(p)} & \pi_2^{(p)} & \cdots & \pi_k^{(p)} \end{bmatrix}$$

where $\pi_n^{(m)} = u_{L,n}^{(m)}$ for $m = 1, \ldots, p$ and we define $\vec{\pi}_n$ to be the n^{th}, p-dimensional column vector of $\mathbf{Z}$. Finally, we define $\mathbf{g}_{N+M} = [\vec{g}_1, \ldots, \vec{g}_N, \vec{g}_{N+1}, \ldots, \vec{g}_{N+M}]^{\top} \in \mathbb{R}^{(N+M)\times p}$ to be a multivariate time series of length $N+M$ where the first N terms are close to $\{\vec{\tilde{y}}_i\}_{i=1}^{N}$ and the remaining M elements will be determined in the following.

VMSSA Recurrent Forecasting

Similar to the univariate case, we use the VMSSA recurrent forecasting algorithm to predict the next term in the reconstructed multivariate time series from the previous $L-1$ terms. Define $\|\cdot\|_F$ to be the Frobenius norm of a $p \times p$ matrix. Now we assume that $\|\mathbf{Z}\mathbf{Z}^{\top}\|_F^2 < 1$ such that $\left(\mathbf{I}_{p\times p} - \mathbf{Z}\mathbf{Z}^{\top}\right)^{-1}$ exists where $\mathbf{I}_{p\times p}$ is the $p \times p$ identity matrix and we define the VMSSA recurrent forecasting algorithm in the following.

Algorithm A - VMSSA Recurrent Forecasting Algorithm:

$$\vec{g}_i = \begin{cases} \vec{\tilde{y}}_i, & i = 1, \ldots, N \\ \sum_{j=1}^{L-1} \mathbf{A}_j \vec{g}_{i+j-L}, & i = N+1, \ldots, N+M \end{cases} \tag{12}$$

where $\mathbf{A}_j = \sum_{n=1}^{k} \left(\mathbf{I}_{p\times p} - \mathbf{Z}\mathbf{Z}^\top\right)^{-1} \vec{\pi}_n \vec{u}_{j,n}^\top$ is a $p \times p$ matrix. For brevity we call this method the *VMSSA R-forecasting* algorithm. Now we discuss the VMSSA vector forecasting approach.

VMSSA Vector Forecasting

For the vector forecasting case, our goal is to find a linear operator $\mathbf{V} : \mathbb{L} \to \mathbb{L}$ such that for some pL-dimensional vector, $\mathbf{y}$, defined by $\mathbf{y} = \mathbf{V}\mathbf{x}$, we have that the distance between the $p(L-1)$-dimensional vectors, $\mathbf{y}^\nabla$ and $\mathbf{x}^\Delta$, is minimal where $\mathbf{x} \in \mathbb{L}$ is a pL-dimensional vector. This is optimally done by way of orthogonal projection. We define the $p(L-1)\times k$ matrix, $\mathbf{U}^\nabla = \left[\mathbf{u}_1^\nabla \cdots \mathbf{u}_k^\nabla\right]$, and now we define the $p(L-1) \times p(L-1)$ orthogonal projection matrix that projects vectors in $\mathbb{R}^{p(L-1)}$ onto $\mathbb{L}^\nabla$ as

$$\mathbf{\Pi} = \mathbf{U}^\nabla \left(\mathbf{U}^\nabla\right)^\top + \mathbf{U}^\nabla \mathbf{Z}^\top \left(\mathbf{I}_{p\times p} - \mathbf{Z}\mathbf{Z}^\top\right)^{-1} \mathbf{Z} \left(\mathbf{U}^\nabla\right)^\top.$$

We may rewrite $\mathbf{\Pi}$ as $\mathbf{\Pi} = \begin{bmatrix} \mathbf{\Pi}^{(1)} \\ \vdots \\ \mathbf{\Pi}^{(p)} \end{bmatrix}$ where the matrix, $\mathbf{\Pi}^{(j)} \in \mathbb{R}^{L-1\times p(L-1)}$, is the projection for the j^{th} variable. Now we define the $p(L-1) \times p$ matrix,

$$\mathcal{R} = \mathbf{U}^\nabla \mathbf{Z}^\top \left(\mathbf{I}_{p\times p} - \mathbf{Z}\mathbf{Z}^\top\right)^{-1} = \left[\mathcal{R}^{(1)} \cdots \mathcal{R}^{(p)}\right],$$

where vector $\mathcal{R}^{(j)} \in \mathbb{R}^{p(L-1)}$ is the recurrent forecast for the j^{th} variable. Similar to the univariate case, we apply the recurrent forecast to $\mathbf{x}^\Delta$ instead of $\mathbf{y}^\nabla$. We assume that $\|\mathbf{Z}\mathbf{Z}^\top\|_F < 1$ such and the expression for $\mathbf{V}$ is given by

$$\mathbf{y} = \mathbf{V}\mathbf{x} = \begin{bmatrix} \mathbf{\Pi}^{(1)}\mathbf{x}^{\Delta} \\ \left(\mathcal{R}^{(1)}\right)^{\top}\mathbf{x}^{\Delta} \\ \mathbf{\Pi}^{(2)}\mathbf{x}^{\Delta} \\ \left(\mathcal{R}^{(2)}\right)^{\top}\mathbf{x}^{\Delta} \\ \vdots \\ \mathbf{\Pi}^{(p)}\mathbf{x}^{\Delta} \\ \left(\mathcal{R}^{(p)}\right)^{\top}\mathbf{x}^{\Delta} \end{bmatrix}.$$

We use $\mathbf{V}$ to predict future pL-dimensional column vectors in the matrix $\mathbf{X}_{\mathfrak{s}}$ and the result is used for forecasting as described in the following algorithm.

Algorithm B - VMSSA Vector Forecasting Algorithm:

1. Define the pL-dimensional column vectors given by

$$\mathbf{w}_j = \begin{cases} \mathbf{x}_j^{\mathfrak{s}}, & j = 1, \ldots, K \\ \mathbf{V}\mathbf{w}_{j-1}, & j = K+1, \ldots, K+M \end{cases}. \tag{13}$$

2. Form the $pL \times (K+M)$ matrix $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{K+M}]$.

3. Perform the block Hankelization routine described in the reconstruction stage of Subsection 1.3.3 on $\mathbf{W}$ in order to extract the multivariate time series $\mathbf{g}_{N+M}$.

4. The terms $\{\vec{g}_{N+1}, \ldots, \vec{g}_{N+M}\}$ form the VMSSA vector forecast.

For brevity we refer to this method as the *VMSSA V-forecasting* algorithm. We see in Chapter 5, that we may avoid having to implement the functional extensions of VMSSA R-forecasting and VMSSA V-forecasting and may instead develop the multivariate functional forecasting algorithms in a manner that is based on tuples of functions.

1.4 Discussion

Thus far, we have given a review of decomposition and prediction techniques for time-independent real-valued and functional data, as well as time series data. As stated before, the goal of this dissertation is to establish rigorous statistical algorithms that can be used to perform nonparametric decomposition and forecasting of time-dependent functional data and we fulfill this goal in the following. We offer the decomposition routine for FTS data, known as functional SSA (FSSA), in

Chapter 2. We find that the FSSA algorithm does a better job in extracting deterministic signals in noisy data as compared to other approaches for simulated and real data. For another avenue in which to view the material presented in Chapter 2, see Haghbin et al. (2021a). From here, we offer the forecasting routines that stem from the FSSA work in Chapter 3. We show that our novel forecasting methodologies outperform current gold standard techniques of FTS prediction for highly periodic simulated and real data. Another way to view the material presented in Chapter 3 is in Trinka et al. (2021). Then we give the multivariate FSSA (MFSSA) over different dimensional domains algorithm in Chapter 4 which is a highly flexible decomposition routine that allows variables to take on many different forms such as curves or images. We also show that the joint multivariate analysis does better in signal extraction of a multivariate FTS (MFTS) signal as compared to FSSA applied to functional variables independently. The work of Chapter 4 can be found in another way in Trinka et al. (2020). Then we develop the forecasting algorithms that stem from the MFSSA work in Chapter 5. We give a real data example that showcases how the multivariate forecasting methods do better than the FSSA forecasting routines for MFTS data. We finish with concluding remarks in Chapter 6 and we offer other related works we've done in the appendices at the end.

CHAPTER 2: FUNCTIONAL SINGULAR SPECTRUM ANALYSIS

In this chapter, we develop a new extension of SSA called FSSA to analyze FTS. The new methodology is constructed by integrating ideas from univariate SSA and FDA. Specifically, we introduce a trajectory operator in the Hilbert space world, which is equivalent to the trajectory matrix in the regular SSA. In the regular SSA, we need to obtain an SVD of the trajectory matrix to decompose a given time series. Since there is no procedure to extract a functional SVD (FSVD) of the trajectory operator, we introduce a computationally tractable algorithm to obtain the FSVD components. The effectiveness of the proposed approach is illustrated by an interesting simulation study example of remote sensing data. Also, we develop a user-friendly R package and a shiny web application to allow for interactive exploration of the results.

2.1 Introduction

In the regular SSA, we assume that the observation at each time point is a scalar. As a matter of interest, we may consider a series of curves observed over time, and use the basics of Hilbert spaces in the FDA framework to introduce the concept of FSSA. In this chapter, we introduce the FSSA algorithm for a time series of functions, where each function can be represented using basis expansion models (e.g. B-spline, or Fourier).

While the research in FDA has grown extensively in recent years, there have been relatively few contributions dealing with FTS; see, e.g., Hörmann and Kokoszka (2012) and Bosq (2000). Although most of the current FTS approaches focus on a parametric fit for inferences and forecasting, there exists other approaches that extend functional principal component analysis to incorporate the temporal correlation of FTS. For instance, Hörmann et al. (2015) introduced dynamic functional principal component analysis (DFPCA) to analyze FTS. This approach makes the strong assumption of stationarity which is usually violated in practice. It would be advantageous to nonparametrically decompose a nonstationary FTS to reveal the respective trends plus seasonal and irregular components in an appropriate manner. As a first step, Fraiman et al. (2014) introduced a new concept of trends for FTS. Hörmann et al. (2018) considered periodic components for FTS and derived procedures to test for periodicity using analysis of frequencies. To the best of our knowledge, existing studies focus on detecting rather than extracting interpretable components.

Since one of the primary missions of SSA is to extract trends and periodic components of a regular (non-functional) time series, it would be rational to establish a similar, elegant, nonparametric

procedure to extract such components for FTS. In this chapter we use the basics of univariate SSA and ideas from FDA to develop the FSSA procedure. In a nutshell, we develop a matrix operator with function entities and provide a procedure to obtain an FSVD of the matrix operator. The process of finding an FSVD for the matrix operator can be seen as a generalization of the regular SVD process used in the SSA literature. The new methodology, FSSA, not only can serve as a nonparametric decomposition tool for FTS; it can also be used as a visualization tool to illustrate the concept of seasonality and periodicity in function space over time.

To illustrate our approach and its utility, consider the following motivating example of intraday call center data. This dataset provides the intraday number of calls to a bank's call center in Israel, during different times of the days for one year. The associated 365 curves are represented in an overlapping pattern in Figure 1 (left).

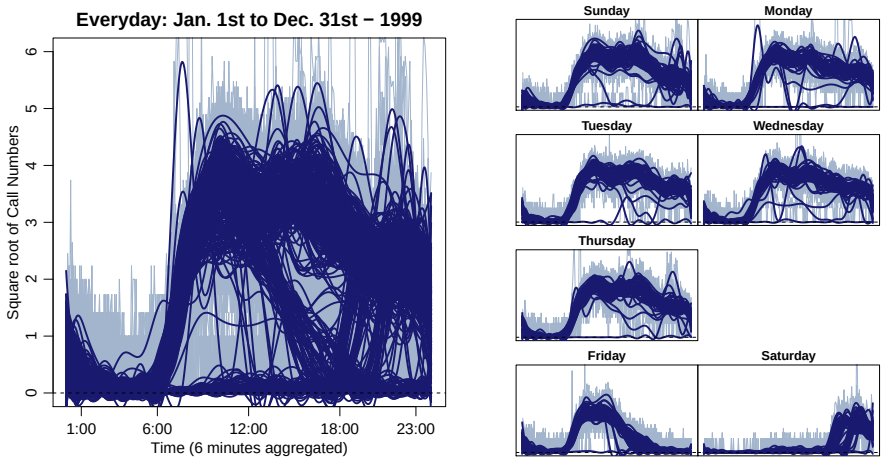

Figure 1: (left): The functional curves that give the number of calls to a bank's call center in Israel between January 1st to December 31st in the year 1999; (right): The same functional observations seen in the plot on the left, partitioned by weekday.

In Figure 1 (right) we investigate the pattern among workdays and weekend days. As we can see, the intraday patterns of weekends (Friday and Saturday) are different from workdays while workdays seem to have similar patterns. Investigators used variants of functional principal component analysis to analyze the call center data in literature (Huang et al., 2008; Maadooliat et al., 2015; Shen and Huang, 2005). For illustrative purposes, we compare the results of the proposed FSSA method, DFPCA, and MSSA applied to the call center dataset. Figure 2 (top) presents the projection of the data onto the first four functional principal components generated from the DFPCA method, which is avaiable for use from the **freqdom.fda** package in R (Hörmann et al., 2015).

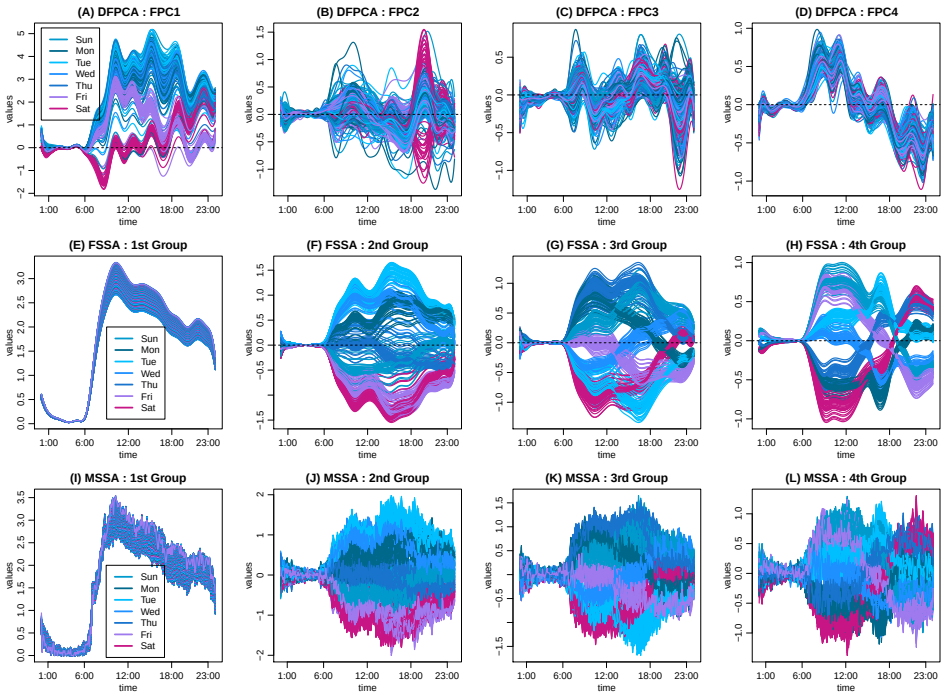

Figure 2: (top): The projection of the call center functional observations onto the four leading functional principal components (FPC)'s generated from DFPCA; (middle): The reconstructed FTS after grouping for FSSA; (bottom): The reconstructed multivariate time series after grouping for MSSA; Seven different colors are used to differentiate between different days of a week.

We used seven different colors to differentiate between different days of a week. We observe that there is no clear separation in either one of the functional principal component graphs in the top row. This may not be surprising, as the purpose of principal component analysis, of any type, is to reduce the dimensionality, and not necessarily decompose the data into regular trends, periodic and irregular components. In contrast, the grouping results that we obtained using the FSSA method on the call center data are given in Figure 2 (middle). We also present the results of MSSA, obtained from the **Rssa** package (Golyandina et al., 2015), in Figure 2 (bottom). It can be seen that the functional behavior of seven days of a week can be well-distinguished, visually, using either one of the last two groups (groups 3 and 4) of the FSSA routine. We also see in Figure 2 (bottom) that it is difficult to distinguish, visually, the seven days of a week using any of the reconstructed multivariate time series generated from MSSA.

The rest of the chapter is organized as follows. Section 2.2 presents the theoretical foundations and some properties of the proposed method (FSSA), and Section 2.3 provides implementation details. Subsection 2.4.1 reports simulation results to illustrate the use of the proposed approach in analyzing FTS, and to compare it with MSSA and DFPCA. Application to a real data example on remote

sensing is given in Subsection 2.4.2. Section 2.5 provides discussion and concluding remarks.

2.2 Theoretical Foundations of Functional Singular Spectrum Analysis

We start this section with the mathematical foundations that are used to develop the FSSA procedure. From hereafter, we consider $\mathbf{y}_N = (y_1, \ldots, y_N)$ as a FTS of length N such that each function vector, $y_i : [0,1] \to \mathbb{R}$, belongs to the Hilbert space $\mathbb{H} = L^2([0,1])$. We define the inner product equipped to $\mathbb{H}$ as $\langle x, y\rangle_{\mathbb{H}} = \int_0^1 x(s)y(s)ds$, where $x, y \in \mathbb{H}$. For a positive integer k, the Hilbert space, $\mathbb{H}^k$, is defined as the Cartesian product of k copies of $\mathbb{H}$; i.e., a length k function vector, $\boldsymbol{x} \in \mathbb{H}^k$, evaluated at a point $\mathbf{s} = [s_1, s_2, \ldots, s_k]^\top \in [0,1]^k$ is a vector of length k denoted by $\boldsymbol{x}(\mathbf{s}) = [x_1(s_1), x_2(s_2), \ldots, x_k(s_k)]^\top$, where $x_i \in \mathbb{H}$. We define the inner product equipped to $\mathbb{H}^k$ as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathbb{H}^k} = \sum_{i=1}^k \langle x_i, y_i\rangle_{\mathbb{H}}$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}^k$. The norms are denoted by $\|\cdot\|_{\mathbb{H}}$ and $\|\cdot\|_{\mathbb{H}^k}$ in the spaces $\mathbb{H}$ and $\mathbb{H}^k$, respectively. For function vectors $x \in \mathbb{H}_1$, and $y \in \mathbb{H}_2$, where $\mathbb{H}_1$ and $\mathbb{H}_2$ are two Hilbert spaces, we define the tensor (outer) product corresponding to the operator $x \otimes y : \mathbb{H}_1 \to \mathbb{H}_2$, as $(x \otimes y)h = \langle x, h\rangle_{\mathbb{H}_1} y$, where $h \in \mathbb{H}_1$ and $\langle\cdot,\cdot\rangle_{\mathbb{H}_1}$ is the inner product equipped to $\mathbb{H}_1$.

For positive integers L and K, we denote $\mathbb{H}^{L\times K}$ as the linear space spanned by operators $\boldsymbol{\mathcal{Z}} : \mathbb{R}^K \to \mathbb{H}^L$, specified by function vectors $(z_{i,j})_{i=1,\ldots,L}^{j=1,\ldots,K}$, where

$$\boldsymbol{\mathcal{Z}}\mathbf{a} = \left(\sum_{j=1}^K a_j z_{1,j}, \ldots, \sum_{j=1}^K a_j z_{L,j}\right), \ z_{i,j} \in \mathbb{H}, \text{ and } \mathbf{a} = [a_1, \ldots, a_K]^\top \in \mathbb{R}^K.$$

We call an operator $\widetilde{\boldsymbol{\mathcal{Z}}} = (\tilde{z}_{i,j}) \in \mathbb{H}^{L\times K}$ Hankel if for all $i = 1, \ldots, L$, and $j = 1, \ldots, K$, we have $\|\tilde{z}_{i,j} - g_s\|_{\mathbb{H}} = 0$ for some $g_s \in \mathbb{H}$ where $s = i + j$. The space of such Hankel operators is denoted by $\mathbb{H}_H^{L\times K}$. For two given operators $\boldsymbol{\mathcal{Z}}_1 = \left(z_{i,j}^{(1)}\right)$ and $\boldsymbol{\mathcal{Z}}_2 = \left(z_{i,j}^{(2)}\right)$ in $\mathbb{H}^{L\times K}$, define

$$\langle \boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2\rangle_F = \sum_{i=1}^L \sum_{j=1}^K \langle z_{i,j}^{(1)}, z_{i,j}^{(2)}\rangle_{\mathbb{H}}.$$

It follows immediately that $\langle\cdot,\cdot\rangle_F$, defines an inner product on $\mathbb{H}^{L\times K}$. We call $\langle\cdot,\cdot\rangle_F$ as the Frobenius inner product of two operators in $\mathbb{H}^{L\times K}$. The associated Frobenius norm is $\|\boldsymbol{\mathcal{Z}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{Z}}\rangle_F}$. Before discussing the FSSA algorithm, we present a lemma that is used in the last step of the proposed algorithm.

Lemma 2.1. *Let $x_i \in \mathbb{H}$ for $i = 1, \ldots, N$. If $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, then*

$$\sum_{i=1}^{N} \|x_i - \bar{x}\|_{\mathbb{H}}^2 \leq \sum_{i=1}^{N} \|x_i - y\|_{\mathbb{H}}^2,$$

for all $y \in \mathbb{H}$.

Proof. We add and subtract $\bar{x}$ to obtain

$$\sum_{i=1}^{N} \|x_i - y\|_{\mathbb{H}}^2 = \sum_{i=1}^{N} \|x_i - \bar{x}\|_{\mathbb{H}}^2 + 2\sum_{i=1}^{N} \langle x_i - \bar{x}, \bar{x} - y\rangle_{\mathbb{H}} + N\|\bar{x} - y\|_{\mathbb{H}}^2.$$

Notice that $\sum_{i=1}^{N}\langle x_i - \bar{x}, \bar{x} - y\rangle_{\mathbb{H}}=0$, then we have

$$\sum_{i=1}^{N} \|x_i - y\|_{\mathbb{H}}^2 = \sum_{i=1}^{N} \|x_i - \bar{x}\|_{\mathbb{H}}^2 + N\|\bar{x} - y\|_{\mathbb{H}}^2 \geq \sum_{i=1}^{N} \|x_i - \bar{x}\|_{\mathbb{H}}^2.$$

□

2.2.1 Functional Singular Spectrum Analysis Algorithm

For an integer $1 < L < N/2$, let $K = N - L + 1$ and define

$$\boldsymbol{x}_j = \Big(y_j, y_{j+1}, \ldots, y_{j+L-1}\Big) \tag{14}$$

to be the j^{th} L-lagged function vector in $\mathbb{H}^L$ corresponding to FTS, $\mathbf{y}_N$, for $j = 1, \ldots, K$. The following provides the FSSA procedure where the FSVD technique is applied to an operator whose range is defined by the linear span of the $\boldsymbol{x}_j$'s where such L-lagged function vectors capture FTS behavior over sub-intervals of time.

Step 1. Embedding

Define the operator $\boldsymbol{\mathcal{X}} : \mathbb{R}^K \to \mathbb{H}^L$ with

$$\boldsymbol{\mathcal{X}}\mathbf{a} = \sum_{j=1}^{K} a_j \boldsymbol{x}_j, \qquad \mathbf{a} = [a_1, \ldots, a_K]^\top \in \mathbb{R}^K. \tag{15}$$

We call $\boldsymbol{\mathcal{X}}$ the trajectory operator. It is easy to see that $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}}\mathbf{y}_N$, where $\boldsymbol{\mathcal{T}}$ is an invertible operator from $\mathbb{H}^N$ to $\mathbb{H}_H^{L\times K}$. Evaluating $\boldsymbol{\mathcal{X}}\mathbf{a}$ at a given L-dimensional vector $\mathbf{s} \in [0,1]^K$ is the same

as the matrix product $\mathbf{X}(\mathbf{s})\mathbf{a}$, where $\mathbf{X}(\mathbf{s})$ is an $L \times K$ Hankel matrix given by

$$\mathbf{X}(\mathbf{s}) = [\boldsymbol{x}_1(\mathbf{s}) \cdots \boldsymbol{x}_K(\mathbf{s})] .$$

Now we define the adjoint of $\boldsymbol{\mathcal{X}}$ in the following proposition.

Proposition 2.2. *The operator, $\boldsymbol{\mathcal{X}}$, is a bounded linear operator. If we define the operator $\boldsymbol{\mathcal{X}}^* : \mathbb{H}^L \to \mathbb{R}^K$, given by the vector*

$$\boldsymbol{\mathcal{X}}^* \boldsymbol{z} = [\langle \boldsymbol{x}_1, \boldsymbol{z} \rangle_{\mathbb{H}^L}, \langle \boldsymbol{x}_2, \boldsymbol{z} \rangle_{\mathbb{H}^L}, \ldots, \langle \boldsymbol{x}_K, \boldsymbol{z} \rangle_{\mathbb{H}^L}]^\top , \ \boldsymbol{z} \in \mathbb{H}^L, \tag{16}$$

then $\boldsymbol{\mathcal{X}}^*$ is the adjoint operator for $\boldsymbol{\mathcal{X}}$.

Proof. The boundedness and linearity of $\boldsymbol{\mathcal{X}}$ is straightforward from the definition of the operator. Now we show the form of the adjoint of $\boldsymbol{\mathcal{X}}$. For a given vector, $\mathbf{a} \in \mathbb{R}^K$, and length L function vector, $\boldsymbol{z} \in \mathbb{H}^L$, we have

$$\begin{aligned} \langle \boldsymbol{\mathcal{X}} \mathbf{a}, \boldsymbol{z} \rangle_{\mathbb{H}^L} &= \sum_{j=1}^{K} a_j \langle \boldsymbol{x}_j, \boldsymbol{z} \rangle_{\mathbb{H}^L} = \sum_{i=1}^{L} \sum_{j=1}^{K} a_j \langle y_{i+j-1}, z_i \rangle_{\mathbb{H}} \\ &= \left\langle \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}, \begin{bmatrix} \sum_{i=1}^{L} \langle y_i, z_i \rangle_{\mathbb{H}} \\ \vdots \\ \sum_{i=1}^{L} \langle y_{i+K-1}, z_i \rangle_{\mathbb{H}} \end{bmatrix} \right\rangle_{\mathbb{R}^K} = \langle \mathbf{a}, \boldsymbol{\mathcal{X}}^* \boldsymbol{z} \rangle_{\mathbb{R}^K}. \end{aligned}$$

□

From here, we enter into step two with the goal of extracting out time-dependent modes of variation from $\boldsymbol{\mathcal{X}}$.

Step 2. Decomposition

In this step, we decompose the trajectory operator, $\boldsymbol{\mathcal{X}}$, into a set of rank one operators. To this end, we denote the range of $\boldsymbol{\mathcal{X}}$ by $R(\boldsymbol{\mathcal{X}})$. Clearly, $R(\boldsymbol{\mathcal{X}}) = \text{span}\,\{\boldsymbol{x}_j\}_{j=1}^{K}$ is $r-$dimensional, where r is a positive integer less than or equal to K. Therefore $\boldsymbol{\mathcal{X}}$ is a finite-rank (r-dimensional) operator. The following theorem provides us with an FSVD of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^*$.

Theorem 2.3. *Consider the trajectory operator, $\boldsymbol{\mathcal{X}}$, in equation* (15)*. There exists orthornormal*

elements, $\{\boldsymbol{\psi}_i\}_{i=1}^r$*, from* $\mathbb{H}^L$ *and orthonormal vectors,* $\{\mathbf{v}_i\}_{i=1}^r$*, from* $\mathbb{R}^K$ *such that*

$$\boldsymbol{\mathcal{X}}\mathbf{a} = \sum_{i=1}^{r} \sqrt{\lambda_i}\langle \mathbf{v}_i, \mathbf{a}\rangle_{\mathbb{R}^K}\boldsymbol{\psi}_i, \quad \textit{for all } \mathbf{a} \in \mathbb{R}^K, \tag{17}$$

where the λ_i*'s are non-ascending positive scalars. We also obtain*

$$\boldsymbol{\mathcal{X}}^*\boldsymbol{z} = \sum_{i=1}^{r} \sqrt{\lambda_i}\langle \boldsymbol{\psi}_i, \boldsymbol{z}\rangle_{\mathbb{H}^L}\mathbf{v}_i, \quad \textit{for all } \boldsymbol{z} \in \mathbb{H}^L. \tag{18}$$

Proof. Since $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^*$ are finite-rank operators, they are also compact and as such, the proof immediately follows from Theorem 7.6 of Weidmann (1980). □

We call $\sqrt{\lambda_i}$ as the i^{th} singular value, $\boldsymbol{\psi}_i$ as the i^{th} left singular function, and $\mathbf{v}_i$ as the i^{th} right singular vector of the trajectory operator, $\boldsymbol{\mathcal{X}}$. The following proposition expands further on the properties of the elements found in an FSVD of $\boldsymbol{\mathcal{X}}$.

Proposition 2.4. *In Theorem 2.3, the set,* $\{\boldsymbol{\psi}_i\}_{i=1}^r$*, forms a basis system for* $R(\boldsymbol{\mathcal{X}})$*, and each vector,* $\mathbf{v}_i$*, can be written as*

$$\mathbf{v}_i = \frac{\boldsymbol{\mathcal{X}}^*\boldsymbol{\psi}_i}{\sqrt{\lambda_i}} = \left[\frac{\langle \boldsymbol{\psi}_i, \boldsymbol{x}_1\rangle_{\mathbb{H}^L}}{\sqrt{\lambda_i}}, \ldots, \frac{\langle \boldsymbol{\psi}_i, \boldsymbol{x}_K\rangle_{\mathbb{H}^L}}{\sqrt{\lambda_i}}\right]^\top, \qquad i = 1, \ldots, K. \tag{19}$$

Proof. We first derive the form of the right singular vectors and then we show that the left singular functions span $R(\boldsymbol{\mathcal{X}})$. We use equation (17) and equation (18) to obtain $\boldsymbol{\mathcal{X}}\mathbf{v}_i = \sqrt{\lambda_i}\boldsymbol{\psi}_i$ and $\boldsymbol{\mathcal{X}}^*\boldsymbol{\psi}_i = \sqrt{\lambda_i}\mathbf{v}_i$. We use equation (17), equation (18), and equation (16) to obtain equation (19). Now by substituting equation (19) into equation (17), we obtain

$$\boldsymbol{x}_j = \sum_{i=1}^{r} \langle \boldsymbol{x}_j, \boldsymbol{\psi}_i\rangle_{\mathbb{H}^L}\boldsymbol{\psi}_i.$$

Since $R(\boldsymbol{\mathcal{X}}) = \operatorname{span}\{\boldsymbol{x}_j\}_{j=1}^K$, it follows that $\{\boldsymbol{\psi}_i\}_{i=1}^r$ is an orthonormal basis for $R(\boldsymbol{\mathcal{X}})$. □

The collection $\{\sqrt{\lambda_i}, \boldsymbol{\psi}_i, \mathbf{v}_i\}$ is defined as the i^{th} eigentriple of $\boldsymbol{\mathcal{X}}$, and the right singular vectors, $\mathbf{v}_i$'s, can be used to produce paired plots similar to paired plots seen in SSA literature (Golyandina et al., 2001). Let $\boldsymbol{\mathcal{X}}_i : \mathbb{R}^K \to \mathbb{H}^L$ be the rank one operator defined by $\boldsymbol{\mathcal{X}}_i = \sqrt{\lambda_i}\mathbf{v}_i \otimes \boldsymbol{\psi}_i$, for $i = 1, \ldots, r$. We use equation (17) and decompose $\boldsymbol{\mathcal{X}}$ as

$$\boldsymbol{\mathcal{X}} = \sum_{i=1}^{r} \boldsymbol{\mathcal{X}}_i. \tag{20}$$

Similar to SSA, we group eigentriples together in disjoint sets to separate out sources of variation and then we add the corresponding rank one operators together within groups. This routine is given in the following step.

Step 3. Grouping

The grouping step consists of rearranging and partitioning the elementary operators $\boldsymbol{\mathcal{X}}_i$'s in equation (20). To do this, we mimic the approaches used in the grouping step of Subsection 1.3.1 for the univariate SSA and implement the equivalent function version of those in Haghbin et al. (2019). We consider a partition, $\{I_1, I_2, \ldots, I_m\}$ of the set of indices, $\{1, \ldots, r\}$, and we define the operator, $\boldsymbol{\mathcal{X}}_{I_q} : \mathbb{R}^K \to \mathbb{H}^L$, given by $\boldsymbol{\mathcal{X}}_{I_q} = \sum_{i \in I_q} \boldsymbol{\mathcal{X}}_i$, such that we have the expansion

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{I_1} + \boldsymbol{\mathcal{X}}_{I_2} + \cdots + \boldsymbol{\mathcal{X}}_{I_m}. \tag{21}$$

The grouping should be done so that each operator, $\boldsymbol{\mathcal{X}}_{I_q} \in \mathbb{H}^{L \times K}$, captures a unique component of variation present in the original FTS such as mean, periodic, or trend behaviors for $q = 1, \ldots, m$. In step four, we have the goal of extracting out a FTS that corresponds with each $\boldsymbol{\mathcal{X}}_{I_q}$.

Step 4. Reconstruction

We use the operator $\boldsymbol{\mathcal{T}}^{-1} : \mathbb{H}_H^{L \times K} \to \mathbb{H}^N$ to transform back each operator, $\boldsymbol{\mathcal{X}}_{I_q}$, seen in equation (21), to a FTS, $\tilde{\mathbf{y}}_N^q$. Since $\boldsymbol{\mathcal{X}}_{I_q} \in \mathbb{H}^{L \times K}$ is not necessarily Hankel, we first project $\boldsymbol{\mathcal{X}}_{I_q}$ to $\mathbb{H}_H^{L \times K}$. Note that $\mathbb{H}_H^{L \times K}$ is a closed subspace of $\mathbb{H}^{L \times K}$, then by Theorem 3.2.3 of Shalit (2017), there exists a unique operator, $\widetilde{\boldsymbol{\mathcal{X}}}_{I_q} \in \mathbb{H}_H^{L \times K}$, such that

$$\|\boldsymbol{\mathcal{X}}_{I_q} - \widetilde{\boldsymbol{\mathcal{X}}}_{I_q}\|_F^2 \leq \|\boldsymbol{\mathcal{X}}_{I_q} - \widetilde{\boldsymbol{\mathcal{Z}}}\|_F^2, \text{ for any operator } \widetilde{\boldsymbol{\mathcal{Z}}} \in \mathbb{H}_H^{L \times K}.$$

To specify $\widetilde{\boldsymbol{\mathcal{X}}}_{I_q}$, we denote the elements of $\boldsymbol{\mathcal{X}}_{I_q}$ and $\widetilde{\boldsymbol{\mathcal{X}}}_{I_q}$ by the function vectors $\left(x_{i,j}^q\right)$ and $\left(\tilde{x}_{i,j}^q\right)$, respectively. Using Lemma 2.1, it is easy to extend the diagonal averaging approach in Golyandina et al. (2001) to $\mathbb{H}^{L \times K}$ and obtain $\tilde{x}_{i,j}^q$'s in the following:

$$\tilde{x}_{i,j}^q = \frac{1}{n_s} \sum_{(l,k): l+k=s} x_{l,k}^q, \tag{22}$$

where $s = i + j$ and n_s stands for the number of (l, k) pairs such that $l + k = s$. This diagonal averaging technique may be viewed as an orthogonal projection operator, $\mathbf{\Pi}_H : \mathbb{H}^{L\times K} \to \mathbb{H}_H^{L\times K}$, and we find that $\widetilde{\boldsymbol{\mathcal{X}}}_{I_q} = \mathbf{\Pi}_H \boldsymbol{\mathcal{X}}_{I_q}$. Now we define $\tilde{\mathbf{y}}_N^q = \boldsymbol{\mathcal{T}}^{-1}\widetilde{\boldsymbol{\mathcal{X}}}_{I_q}$ as a reconstructed FTS. Similar to SSA, the goal of FSSA is to separate out an FTS into additive components. In the following we introduce the notion of separability in the function space realm.

2.2.2 Separability

Let $\mathbf{y}_N = \mathbf{y}_N^{(1)} + \mathbf{y}_N^{(2)}$, where $\mathbf{y}_N^{(i)} = \left(y_1^{(i)}, \ldots, y_N^{(i)}\right)$, $i = 1, 2$, are FTS of length N. Using a fixed window length L, for each series $\mathbf{y}_N^{(i)}$, denote $\{\boldsymbol{x}_k^{(i)}\}_{k=1}^K$ as a sequence of L-lagged function vectors corresponding to $\mathbf{y}_N^{(i)}$, and $L^{(i)}$ as the linear space spanned by $\{\boldsymbol{x}_k^{(i)}\}_{k=1}^K$. Analogous to univariate SSA, separability of the series $\mathbf{y}_N^{(1)}$ and $\mathbf{y}_N^{(2)}$ is equivalent to $L^{(1)} \perp L^{(2)}$, which is the same as $\langle \boldsymbol{x}_k^{(1)}, \boldsymbol{x}_{k'}^{(2)} \rangle_{\mathbb{H}^L} = 0$, for all $k, k' = 1, \ldots, K$. Furthermore, a necessary condition for separability can be defined based on the w-correlation measure. To do this, consider the weighted inner product of two FTS, $\mathbf{y}_N^{(1)}$ and $\mathbf{y}_N^{(2)}$, as

$$\left\langle \mathbf{y}_N^{(1)}, \mathbf{y}_N^{(2)} \right\rangle_w = \sum_{i=1}^N w_i \langle y_i^{(1)}, y_i^{(2)} \rangle_{\mathbb{H}}, \tag{23}$$

where $w_i = \min\{i, L, N - i + 1\}$. We call the series $\mathbf{y}_N^{(1)}$ and $\mathbf{y}_N^{(2)}$ w-orthogonal if

$$\left\langle \mathbf{y}_N^{(1)}, \mathbf{y}_N^{(2)} \right\rangle_w = 0.$$

Theorem 2.5. *If the series $\mathbf{y}_N^{(1)}$ and $\mathbf{y}_N^{(2)}$ are separable, then they are w-orthogonal.*

Proof. Notice that equation (23) gives

$$\begin{aligned}
\left\langle \mathbf{y}_N^{(1)}, \mathbf{y}_N^{(2)} \right\rangle_w &= \sum_{i=1}^N w_i \langle y_i^{(1)}, y_i^{(2)} \rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{L-1} i \langle y_i^{(1)}, y_i^{(2)} \rangle_{\mathbb{H}} + \sum_{i=L}^{K} L \langle y_i^{(1)}, y_i^{(2)} \rangle_{\mathbb{H}} + \sum_{i=K+1}^{N} (N - i + 1) \langle y_i^{(1)}, y_i^{(2)} \rangle_{\mathbb{H}} \\
&= \sum_{s=2}^{L} (s-1) \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} + \sum_{s=L+1}^{K+1} L \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} + \sum_{i=K+2}^{N+1} (N - s + 2) \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}}. \\
&= \sum_{s=2}^{L} \sum_{j=1}^{s-1} \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} + \sum_{s=L+1}^{K+1} \sum_{j=s-L}^{s-1} \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} + \sum_{i=K+2}^{K+L} \sum_{j=s-L}^{K} \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} \\
&= \sum_{k=1}^{K} \sum_{s=k+1}^{L+k} \langle y_{s-1}^{(1)}, y_{s-1}^{(2)} \rangle_{\mathbb{H}} = \sum_{k=1}^{K} \sum_{i=1}^{L} \langle y_{i+k-1}^{(1)}, y_{i+k-1}^{(2)} \rangle_{\mathbb{H}} = \sum_{k=1}^{K} \langle \boldsymbol{x}_k^{(1)}, \boldsymbol{x}_k^{(2)} \rangle_{\mathbb{H}^L}
\end{aligned}$$

Hence, separability of $\mathbf{y}_N^{(1)}$ and $\mathbf{y}_N^{(2)}$ implies $\langle \boldsymbol{x}_k^{(1)}, \boldsymbol{x}_k^{(2)} \rangle_{\mathbb{H}^L} = 0$ for all $k = 1, \ldots, K$, which completes the proof. □

The function version of the w-correlation measure can be obtained by replacing the definition of the weighted inner product given in equation (23) into equation (5). We define the w-correlation matrix in a similar fashion as compared to SSA which can be used in the grouping step of FSSA. For now we continue into the computer implementation strategy of FSSA.

2.3 Computer Implementation Strategy

In the following we define a basis system used to estimate each function vector in $\mathbb{H}$. From here, we develop basis elements used to estimate each L-lagged function vector, $\boldsymbol{x}_j$ in $\mathbb{H}^L$, and the trajectory operator $\boldsymbol{\mathcal{X}}$. We then develop a theorem that implements an FSVD of the estimated trajectory operator.

In practice, functional data are being recorded discretely and then converted to functional objects using proper smoothing techniques. We refer to Ramsay and Silverman (2007) for more details on preprocessing the raw data. Let $\{\nu_i\}_{i\in\mathbb{N}}$ be a known basis system (not necessarily orthogonal) of $\mathbb{H}$. Each functional observation vector in $\mathbb{H}$ can be projected onto the subspace $\mathbb{H}_d = \text{span}\,\{\nu_i\}_{i=1}^d$, where d can be determined by variety of techniques (e.g. cross-validation where we select the value of d that minimizes reconstruction test root mean square error). Therefore, each function vector, $y_j \in \mathbb{H}_d$, is uniquely represented by

$$y_j = \sum_{i=1}^{d} c_{ij}\nu_i, \quad j = 1, \ldots, N.$$

Let us define the quotient sequence, q_k, and the remainder sequence, r_k, by

$$k = (q_k - 1)L + r_k, \qquad 1 \le r_k \le L, \quad 1 \le q_k \le d. \tag{24}$$

Note that for any given k ($1 \le k \le Ld$), one may use equation (24) to determine q_k and r_k uniquely, so these sequences are well-defined. Consider $\boldsymbol{\phi}_k$ as a length L function vector whose entries are all zero functions, except for the r_k-th element, which is ν_{q_k}. The following lemma expands on the useful properties of the $\boldsymbol{\phi}_k$'s.

Lemma 2.6. *The sequence, $\{\boldsymbol{\phi}_k\}_{k=1}^{Ld}$, is a basis system for the subspace, $\mathbb{H}_d^L \subset \mathbb{H}^L$, where $\mathbb{H}_d^L$ is the Cartesian product of L copies of $\mathbb{H}_d$.*

Proof. The proof will be divided into two steps. In the first step it will be shown that $\mathbb{H}_d^L = \text{span}\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{Ld}\}$. In the second step it will be proved that $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{Ld}$ are linearly independent. Let $\boldsymbol{z} = (z_1, \ldots, z_L)$ be a length L function vector in $\mathbb{H}_d^L$. By definition, each component of $\boldsymbol{z}$ admits the basis expansions $z_j = \sum_{i=1}^d b_{i,j}\nu_i,\ j = 1, \ldots, L$. Therefore

$$\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^d b_{i,1}\nu_i(s) \\ \sum_{i=1}^d b_{i,2}\nu_i(s) \\ \vdots \\ \sum_{i=1}^d b_{i,L}\nu_i(s) \end{pmatrix} = b_{1,1}\begin{pmatrix} \nu_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + b_{1,L}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_1 \end{pmatrix} + b_{2,1}\begin{pmatrix} \nu_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$
$$+ \cdots + b_{2,L}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_2 \end{pmatrix} + \cdots + b_{d,1}\begin{pmatrix} \nu_d \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + b_{d,L}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_d \end{pmatrix} = \sum_{k=1}^{Ld} b_{q_k,r_k}\boldsymbol{\phi}_k,$$

which implies the first step. To prove linear independency, if $\sum_{k=1}^{Ld} a_k\boldsymbol{\phi}_k = \mathbf{0}$ then,

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = a_1\begin{pmatrix} \nu_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + a_L\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_1 \end{pmatrix} + a_{L+1}\begin{pmatrix} \nu_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + a_{2L}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_2 \end{pmatrix}$$
$$+ \cdots + a_{(d-1)L+1}\begin{pmatrix} \nu_d \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + a_{dL}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \nu_d \end{pmatrix}$$

$$= \begin{pmatrix} a_1\nu_1 + a_{L+1}\nu_2 + \cdots + a_{(d-1)L+1}\nu_d \\ a_2\nu_1 + a_{L+2}\nu_2 + \cdots + a_{(d-1)L+2}\nu_d \\ \vdots \\ a_L\nu_1 + a_{2L}\nu_2 + \cdots + a_{dL}\nu_d \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^d a_{(j-1)L+1}\nu_j \\ \sum_{j=1}^d a_{(j-1)L+2}\nu_j \\ \vdots \\ \sum_{j=1}^d a_{jL}\nu_j \end{pmatrix}.$$

This means $\sum_{j=1}^d a_{(j-1)L+i}\nu_j = 0$ for all $i = 1, \ldots, L$ and consequently, each scalar, $a_{(j-1)L+i} = 0$, for $j = 1, \ldots, d$, since $\{\nu_i\}_{i=1}^d$ are linearly independent. □

Now using Lemma 2.6 and equation (14), each L-lagged function vector, $\boldsymbol{x}_j \in \mathbb{H}_d^L$, admits a unique representation as

$$\boldsymbol{x}_j = \sum_{i=1}^{Ld} b_{ij}\boldsymbol{\phi}_i = \boldsymbol{\Phi}\mathbf{b}_j \quad j = 1, \ldots, K,$$

where $\boldsymbol{\Phi} : \mathbb{R}^{Ld} \to \mathbb{H}_d^L$ is a linear operator whose range is linearly spanned by $\{\boldsymbol{\phi}_k\}_{k=1}^{Ld}$ and we have the Ld-dimensional vector, $\mathbf{b}_j$, follows the form of

$$\mathbf{b}_j = \left[c_{1j}, \ldots, c_{1,j+L-1}, c_{2j}, \ldots, c_{2,j+L-1}, \ldots, c_{d,j+L-1}\right]^\top.$$

If we define $\boldsymbol{\mathcal{X}} : \mathbb{R}^K \to \mathbb{H}_d^L$ as a bounded linear operator whose range is linearly spanned by $\{\boldsymbol{x}_j\}_{j=1}^K$, then for any $\mathbf{a} \in \mathbb{R}^K$, we have

$$\boldsymbol{\mathcal{X}}\mathbf{a} = \sum_{j=1}^{K} a_j\boldsymbol{x}_j = \sum_{j=1}^{K} a_j\boldsymbol{\Phi}\mathbf{b}_j = \boldsymbol{\Phi}\mathbf{B}\mathbf{a}, \tag{25}$$

where $\mathbf{B} = [b_{i,j}]_{i=1\ldots Ld}^{j=1\ldots K} = [\mathbf{b}_1 \cdots \mathbf{b}_K]_{Ld\times K}$ is a matrix. We now give a theorem that defines the matrix that implements $\boldsymbol{\mathcal{X}}$ and this same theorem gives the technique used to estimate the corresponding FSVD elements.

Theorem 2.7. *Define the $Ld \times K$ matrix, $\mathbf{X} = \mathbf{G}^{1/2}\mathbf{B}$, where $\mathbf{G} = \left[\langle\boldsymbol{\phi}_i, \boldsymbol{\phi}_j\rangle_{\mathbb{H}^L}\right]_{i,j=1}^{Ld}$ is the $Ld \times Ld$ Gram matrix. Let us denote the collection $\{\sqrt{\lambda_i}, \mathbf{u}_i, \mathbf{v}_i\}$ as the ith eigentriple of $\mathbf{X}$. Now define the i^{th} left singular function of $\boldsymbol{\mathcal{X}}$ with $\boldsymbol{\psi}_i = \boldsymbol{\Phi}\mathbf{w}_i$, where vector $\mathbf{w}_i = \mathbf{G}^{-1/2}\mathbf{u}_i$. The following holds:*

i) $\boldsymbol{\mathcal{X}}^*\boldsymbol{\psi}_i = \sqrt{\lambda_i}\mathbf{v}_i$.

ii) $\boldsymbol{\mathcal{X}}\mathbf{v}_i = \sqrt{\lambda_i}\boldsymbol{\psi}_i$.

iii) $\{\boldsymbol{\psi}_i\}_{i=1}^r$ *is an orthonormal basis for* $R(\boldsymbol{\mathcal{X}})$.

Proof. We prove each part of this theorem in succession.

i) For any two length L function vectors, $\boldsymbol{x} = \boldsymbol{\Phi}\mathbf{b}$ and $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{c}$, in $\mathbb{H}_d^L$, it is easy to see that

$$\langle\boldsymbol{x}, \boldsymbol{y}\rangle_{\mathbb{H}^L} = \mathbf{b}^\top\mathbf{G}\mathbf{c}, \quad \mathbf{b}, \mathbf{c} \in \mathbb{R}^{Ld}. \tag{26}$$

Therefore,

$$\begin{aligned}\boldsymbol{\mathcal{X}}^*\boldsymbol{\psi}_i &= \left[\langle \boldsymbol{x}_1, \boldsymbol{\psi}_i\rangle_{\mathbb{H}^L}, \ldots, \langle \boldsymbol{x}_K, \boldsymbol{\psi}_i\rangle_{\mathbb{H}^L}\right]^\top = \left[\mathbf{w}_i^\top \mathbf{G}\mathbf{b}_1, \ldots, \mathbf{w}_i^\top \mathbf{G}\mathbf{b}_k\right]^\top \\ &= \mathbf{B}^\top \mathbf{G}\mathbf{w}_i = \mathbf{B}^\top \mathbf{G}^{1/2}\mathbf{u}_i = \mathbf{X}^\top \mathbf{u}_i = \sqrt{\lambda_i}\mathbf{v}_i.\end{aligned}$$

ii) Using equation (25) gives

$$\begin{aligned}\boldsymbol{\mathcal{X}}\mathbf{v}_i &= \boldsymbol{\Phi}\mathbf{B}\mathbf{v}_i = \boldsymbol{\Phi}\mathbf{G}^{-1/2}\mathbf{G}^{1/2}\mathbf{B}\mathbf{v}_i = \boldsymbol{\Phi}\mathbf{G}^{-1/2}\mathbf{X}\mathbf{v}_i \\ &= \sqrt{\lambda_i}\boldsymbol{\Phi}\mathbf{G}^{-1/2}\mathbf{u}_i = \sqrt{\lambda_i}\boldsymbol{\Phi}\mathbf{w}_i = \sqrt{\lambda_i}\boldsymbol{\psi}_i.\end{aligned}$$

iii) First we prove that the left singular functions, $\boldsymbol{\psi}_i'$s, belong to $R(\boldsymbol{\mathcal{X}})$. From (ii) and equation (15) we have

$$\boldsymbol{\psi}_i = \frac{\boldsymbol{\mathcal{X}}\mathbf{v}_i}{\sqrt{\lambda_i}} = \sum_{j=1}^{K} \frac{v_{ij}}{\sqrt{\lambda_i}}\boldsymbol{x}_j \in R(\boldsymbol{\mathcal{X}}), \qquad \text{where } \mathbf{v}_i = \left[v_{i1}, \ldots, v_{iK}\right]^\top.$$

Next, to show orthonormality, for any $i, j = 1, \ldots, r$ we have

$$\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j\rangle_{\mathbb{H}^L} = \mathbf{w}_i^\top \mathbf{G}\mathbf{w}_j = \mathbf{u}_i^\top \mathbf{u}_j = \delta_{i,j}.$$

We note that $\sum_{i=1}^r \mathbf{u}_i\mathbf{u}_i^\top = \mathbf{I}_r$, where $\mathbf{I}_r$ is the $r \times r$ identity matrix and therefore $\sum_{i=1}^r \mathbf{w}_i\mathbf{w}_i^\top = \mathbf{G}^{-1}$. We use this result to show that any length L function vector, $\boldsymbol{x} \in R(\boldsymbol{\mathcal{X}})$, can be written as a linear combination of $\boldsymbol{\psi}_i$'s. To this end, note that for all $\boldsymbol{x} \in R(\boldsymbol{\mathcal{X}})$, there exists a vector $\mathbf{a} \in \mathbb{R}^K$, such that $\boldsymbol{x} = \boldsymbol{\mathcal{X}}\mathbf{a}$. Using equation (25) and equation (26) we have

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{\Phi}\mathbf{B}\mathbf{a} = \boldsymbol{\Phi}\mathbf{G}^{-1}\mathbf{G}\mathbf{B}\mathbf{a} = \boldsymbol{\Phi}\left(\sum_{i=1}^r \mathbf{w}_i\mathbf{w}_i^\top\right)\mathbf{G}\mathbf{B}\mathbf{a} = \sum_{i=1}^r \boldsymbol{\Phi}\mathbf{w}_i\mathbf{w}_i^\top\mathbf{G}\mathbf{B}\mathbf{a} \\ &= \sum_{i=1}^r \boldsymbol{\psi}_i\mathbf{w}_i^\top\mathbf{G}\mathbf{B}\mathbf{a} = \sum_{i=1}^r \langle \boldsymbol{x}, \boldsymbol{\psi}_i\rangle_{\mathbb{H}^L}\boldsymbol{\psi}_i.\end{aligned}$$

It remains to be proven that the $\boldsymbol{\psi}_i'$s are linearly independent. Note that

$$\sum_{i=1}^r c_i\boldsymbol{\psi}_i = \sum_{i=1}^r c_i\boldsymbol{\Phi}\mathbf{w}_i = \boldsymbol{\Phi}\sum_{i=1}^r c_i\mathbf{G}^{-1/2}\mathbf{u}_i = \boldsymbol{\Phi}\mathbf{G}^{-1/2}\sum_{i=1}^r c_i\mathbf{u}_i.$$

Therefore $\sum_{i=1}^r c_i\boldsymbol{\psi}_i = 0$ implies $\sum_{i=1}^r c_i\mathbf{u}_i = 0$, and the linear independency of $\mathbf{u}_i$'s implies

$c_i = 0$ for all $i = 1, \cdots, r$, which completes the proof.

□

Now, we introduce the following corollary which defines an FSVD of $\boldsymbol{\mathcal{X}}$ in terms of eigentriples.

Corollary 2.8. *The collection of eigentriples,* $\{\sqrt{\lambda_i}, \boldsymbol{\psi}_i, \mathbf{v}_i\},\ i = 1, \ldots, r$ *defined in the Theorem 2.7, defines an FSVD of* $\boldsymbol{\mathcal{X}}$.

Note that the Gram matrix, $\mathbf{G}$, used in Theorem 2.7, can be further simplified to $\mathbf{G} = \left[\delta_{r_i,r_j}\langle \nu_{q_i}, \nu_{q_j}\rangle_{\mathbb{H}}\right]_{i,j=1}^{Ld}$ where $\delta_{r_i,r_j} = 1$ if $r_i = r_j$ and zero otherwise. We now implement our FSSA algorithm in numerical studies designed to show the utility of the method.

2.4 Numerical Study

In Subsection 2.4.1 we present a simulation study to elaborate on the use of the FSSA compared with DFPCA and MSSA under different scenarios. To do so, we utilize the implementation of the proposed FSSA method that is available as an R package named **Rfssa** in the CRAN repository (Haghbin et al., 2019). We also use the **freqdom.fda** (Hörmann et al., 2015) and **Rssa** (Golyandina et al., 2015) packages to obtain the DFPCA and MSSA results. In Subsection 2.4.2, we analyze remote sensing data using **Rfssa** and provide some visualization tools that are useful for the grouping step.

We developed a shiny app, included in the **Rfssa** package, also available at `https://fssa.shinyapps.io/fssa/`, to demonstrate and reproduce different aspects of the simulation setup. Furthermore, the app can be used to compare the results of DFPCA, MSSA, and FSSA on the remote sensing data, call center dataset or any other FTS, provided by the end-user.

2.4.1 Simulation Study

For the simulation setup, consider FTS of lengths $N = 50, 100, 150$ and 200 which are observed in $n = 100$ fixed equidistant discrete points on $[0, 1]$ from the following model:

$$Y_t(s_i) = m_t(s_i) + X_t(s_i),\ s_i \in [0,1], i = 1, \ldots, n, \text{and } t = 1, \ldots, N. \tag{27}$$

A basis of 15 cubic B-spline functions is used to convert the discretized Y_t's into smooth (continuous) function curves. In this model, the $m_t(s)$ function is considered to be a periodic component defined

as

$$m_t(s) = e^{s^2}\cos(2\pi\omega t) + \cos(4\pi s)\sin(2\pi\omega t), \tag{28}$$

where ω is the model frequency with three different values ($\omega = 0, 0.1$ and 0.25). Figure A2 in the Appendices depicts a perspective and a heatmap view of the FTS given in equation (28) for N = 50 and $\omega = 0.1$.

The $X_t(s)$ function in equation (27) is a stochastic term that is generated under four different settings with an increasing trend in complexity. In the first setting, we consider the elements of the set $\{X_t(s_i)|i = 1, \ldots, n \text{ and } t = 1, \ldots, N\}$ to be drawn from an independent Gaussian White Noise (GWN) process with zero mean and standard deviation equal to 0.1. It is expected to obtain an acceptable performance from functional principal component analysis for reconstructing the FTS in the first setting as intuitively, the functional principal component analysis method outperforms other approaches under this ideal framework (see Maadooliat et al., 2015, for more details).

In the remaining three settings, the $X_t(s)$ processes are simulated from a functional autoregressive model of order 1, $FAR(1)$, defined by

$$X_t(s) = \Psi X_{t-1}(s) + \varepsilon_t(s), \tag{29}$$

where Ψ is an integral operator with a parabolic kernel as follows

$$\psi(s,u) = \gamma_0\left(2 - (2s-1)^2 - (2u-1)^2\right). \tag{30}$$

The constant, γ_0, is chosen such that the Hilbert-Schmidt norm defined by

$$\|\Psi\|_{\mathcal{S}}^2 = \int_0^1\int_0^1 |\psi(s,u)|^2 ds du, \tag{31}$$

acquires the values $\|\Psi\|_{\mathcal{S}}^2 = 0, 0.5$, and 0.9, for the remaining three settings, respectively. In these settings, the white noise functions, $\varepsilon_t(s)$, are considered as independent trajectories of standard Brownian motion over the interval, $[0, 1]$. It is worth to note that as we increase the Hilbert-Schmidt norm, $\|\Psi\|_{\mathcal{S}}^2$, in the FAR(1) models, the dependency structure of consecutive FTS increases, and we expect it would be more challenging to reconstruct the true structures.

To compare the performance of FSSA and MSSA, we consider three window length parameters ($L = 20, 30$, and 40) in our simulation setup. Note that we may break the rule of thumb that

$1 < L < N/2$ and still run the FSSA and MSSA algorithms with a lag that is greater than $N/2$ with the goal of extracting out modes of variation from data that have small frequencies (longer periods of oscillation). As such, we allow for the comparison of the SSA-based methodologies in the simulation for when such cases arise (e.g. $N = 50$ and $L = 30$). For the sake of consistency in all of the reconstruction procedures (DFPCA, MSSA and FSSA), we use the first two leading eigen-components. The ratio of amplitude of the signal to error is used as the signal-to-noise ratio $\left(SNR = \frac{\sum_{t=1}^{N}\sum_{i=1}^{n} m_t(s_i)^2}{\sum_{t=1}^{N}\sum_{i=1}^{n} X_t(s_i)^2}\right)$. As a measure of goodness of fit, we use the Root Mean Square Error (RMSE) defined as:

$$RMSE = \sqrt{\frac{1}{N \times n}\sum_{t=1}^{N}\sum_{i=1}^{n}\left(m_t(s_i) - \hat{Y}_t(s_i)\right)^2},$$

where $\hat{Y}_t(s_i)$ is the FTS reconstructed by each method evaluated at a point $s_i \in [0, 1]$. We repeat each setting 1000 times and report the mean of the RMSE's in Table 1.

Model	ω	N	DFPCA	MSSA			FSSA			SNR
				$L=20$	$L=30$	$L=40$	$L=20$	$L=30$	$L=40$	
Setting 1 GWN	0.00	50	0.092	0.026	0.022	0.019	**0.010**	0.011	0.014	241.91
		100	0.088	0.024	0.020	0.018	0.008	**0.007**	**0.007**	241.57
		150	0.074	0.024	0.020	0.017	0.007	**0.006**	**0.006**	240.37
		200	0.071	0.024	0.019	0.017	0.007	**0.006**	**0.006**	241.08
	0.10	50	0.062	0.028	0.024	0.022	**0.009**	0.011	0.014	146.95
		100	0.045	0.027	0.023	0.020	**0.006**	**0.006**	0.007	147.20
		150	0.037	0.027	0.022	0.019	**0.005**	**0.005**	**0.005**	146.23
		200	0.033	0.026	0.022	0.019	**0.005**	**0.005**	**0.005**	146.60
	0.25	50	0.046	0.028	0.024	0.022	**0.009**	0.011	0.014	147.34
		100	0.038	0.027	0.022	0.020	**0.006**	**0.006**	0.007	145.87
		150	0.032	0.027	0.022	0.019	**0.005**	**0.005**	**0.005**	147.35
		200	0.030	0.026	0.022	0.019	**0.005**	**0.005**	**0.005**	147.07
Setting 2 $\|\Psi\|^2_{\mathcal{S}}=0$	0.00	50	1.008	0.251	0.246	0.270	**0.244**	0.248	0.284	7.62
		100	0.993	0.205	0.185	0.179	0.190	0.176	**0.174**	5.73
		150	0.974	0.192	0.166	0.153	0.175	0.155	**0.145**	5.76
		200	0.960	0.186	0.158	0.143	0.168	0.145	**0.134**	5.68
	0.10	50	0.636	0.220	0.211	0.235	**0.204**	0.216	0.264	3.34
		100	0.633	0.193	0.165	0.154	0.158	0.144	**0.143**	4.04
		150	0.633	0.188	0.158	0.141	0.149	0.130	**0.120**	4.39
		200	0.632	0.186	0.154	0.136	0.144	0.123	**0.112**	3.66
	0.25	50	0.636	0.223	0.214	0.235	**0.206**	0.218	0.263	3.44
		100	0.633	0.194	0.167	0.156	0.160	0.146	**0.144**	4.33
		150	0.633	0.188	0.157	0.140	0.148	0.129	**0.120**	3.92
		200	0.632	0.185	0.153	0.136	0.143	0.122	**0.111**	4.25
Setting 3 $\|\Psi\|^2_{\mathcal{S}}=0.5$	0.00	50	0.841	0.316	0.310	0.350	**0.305**	0.311	0.364	7.20
		100	0.818	0.271	0.240	0.231	0.250	0.228	**0.224**	5.36
		150	0.806	0.262	0.223	0.203	0.238	0.207	**0.192**	5.67
		200	0.801	0.258	0.216	0.194	0.233	0.198	**0.180**	5.58
	0.10	50	0.696	0.277	0.267	0.300	**0.255**	0.270	0.330	3.27
		100	0.709	0.241	0.207	0.193	0.198	0.180	**0.178**	4.04
		150	0.710	0.235	0.197	0.176	0.187	0.162	**0.151**	4.23
		200	0.710	0.232	0.192	0.170	0.180	0.153	**0.139**	3.48
	0.25	50	0.707	0.214	0.205	0.225	**0.198**	0.209	0.254	2.93
		100	0.707	0.187	0.160	0.149	0.153	0.139	**0.138**	4.20
		150	0.705	0.181	0.151	0.135	0.141	0.123	**0.114**	3.54
		200	0.705	0.178	0.148	0.130	0.136	0.116	**0.106**	4.12
Setting 4 $\|\Psi\|^2_{\mathcal{S}}=0.9$	0.00	50	0.926	0.486	0.477	0.535	**0.464**	0.476	0.553	6.29
		100	0.915	0.441	0.391	0.373	0.404	0.368	**0.358**	4.70
		150	0.901	0.432	0.371	0.339	0.391	0.341	**0.318**	5.16
		200	0.823	0.432	0.363	0.326	0.388	0.331	**0.301**	5.02
	0.10	50	0.830	0.341	0.330	0.384	**0.313**	0.331	0.413	2.84
		100	0.831	0.286	0.245	0.229	0.235	0.212	**0.210**	3.78
		150	0.832	0.279	0.233	0.208	0.220	0.190	**0.176**	3.63
		200	0.829	0.275	0.227	0.200	0.213	0.179	**0.163**	2.96
	0.25	50	0.829	0.210	0.200	0.226	**0.195**	0.205	0.255	2.28
		100	0.829	0.175	0.149	0.138	0.140	0.129	**0.127**	3.60
		150	0.829	0.169	0.141	0.125	0.129	0.112	**0.105**	2.85
		200	0.833	0.167	0.138	0.121	0.124	0.106	**0.097**	3.40

Table 1: The mean of RMSE for 1000 generations of the simulated model by DFPCA, MSSA and FSSA approaches. The smallest error in each row is shown in bold.

By comparing the results in Table 1, it can be seen that FSSA outperforms DFPCA in different scenarios. This may not be surprising, as the main task of DFPCA is dimension reduction. Except for the first noise setting (GWN), MSSA also outperforms DFPCA. Furthermore, FSSA performs better than MSSA in most of the cases except for where the length of the FTS is small ($N = 50$) and

the window size, L, is getting closer to N. However, the simulation study suggests that FSSA is the optimal method for reconstructing the longer FTS ($N \geq 100$). For all methods, RMSE decreases as the length of the FTS increases. For two smaller frequencies ($\omega = 0$ and 0.1), the average of RMSE increases as $\|\Psi\|_{\mathcal{S}}^2$ increases while it decreases for $\omega = 0.25$.

The ratio of RMSE of MSSA to FSSA is examined in Figure 3 for different window lengths (L), FTS lengths (N) and frequencies (ω).

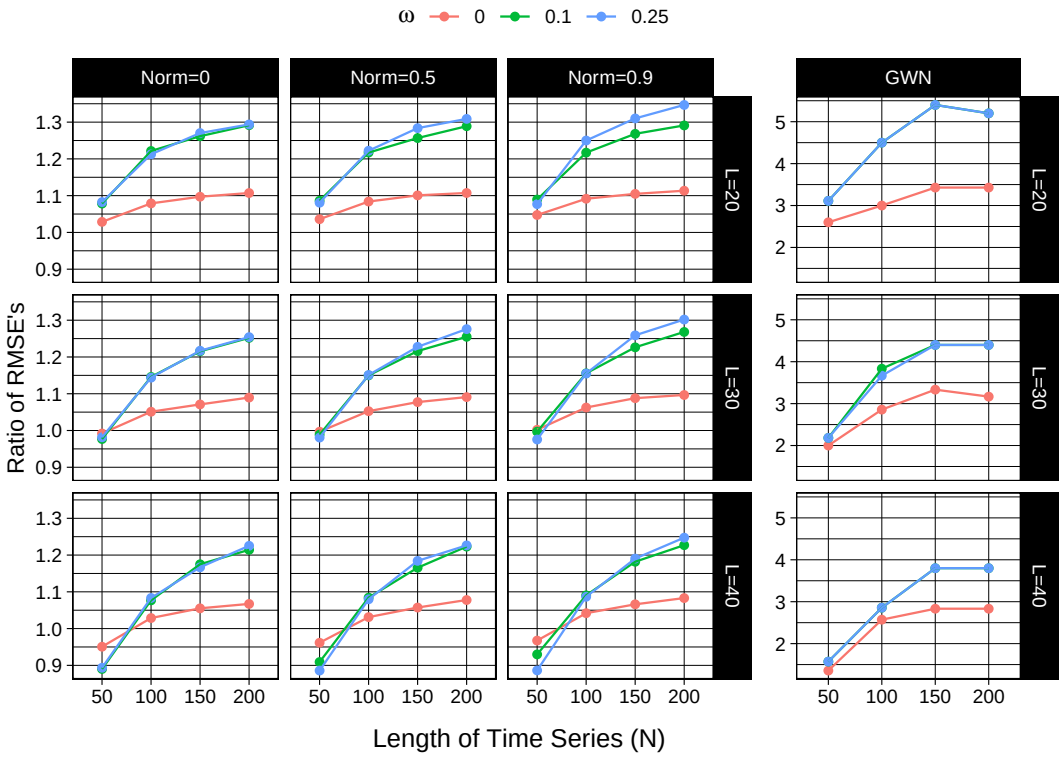

Figure 3: Ratio of RMSE of MSSA to FSSA in simulation study with 1000 repetitions.

We have that high RMSE values for MSSA and low RMSE values for FSSA correspond with a higher ratio and we see that as N increases, FSSA does a better job in extraction of $m_t(s)$ as compared with MSSA. Overall, as L is increasing, the pattern of ratio of RMSE's remains unchanged. Although as the window length becomes larger, either the improvement diminishes for longer FTS, or disappears (or reverses) for smaller N. It is also worth to note that in setting 1 (GWN), based on the right panel of Figure 3 and Table 1, the FSSA approach dominates the other two methods in all combinations of parameters with a better efficiency scale. To view a subset of the simulation setups and results, see Section A.1 in the Appendices.

2.4.2 Application to Remote Sensing Data

Tracking changes in vegetation over time has become of interest to researchers that want to preserve wildlife. One technique that is used to track how much vegetation is present in a region is through field surveys. The problem with the field survey method is that it is difficult to implement especially in low population areas (Panuju and Trisasongko, 2012). The use of remote sensing data, in the context of tracking man-made or natural changes in vegetation, is preferred. One source of remote sensing data is from NASA's MODerate-resolution Imaging Spectroradiometer (MODIS) satellite which provides images, twice daily, of regions around the globe at varying spatial resolutions (Tuck et al., 2014). Normalized Difference Vegetation Index (NDVI) is a commonly used pixel-wise index in MODIS satellite images. The NDVI values are bounded between zero and one, where index values that are closer to zero indicate that less vegetation is present and larger values indicate a higher presence of vegetation (Tuck et al., 2014).

Many studies have used the spectral NDVI measure and its variants in order to remotely track changes in vegetation over time (Lambin, 1999). The temporal average and temporal variability of NDVI images have been used as explanatory variables for the number of different types of vegetation present in many regions (Tuck et al., 2014). Panuju and Trisasongko (2012) used the maximum value of NDVI images, taken of the Jambi province in Indonesia, to form a time series. The resulting time series was then analyzed using an X12-ARIMA model in order to identify trend and seasonal changes in woody vegetation (Panuju and Trisasongko, 2012). While these sample statistics (e.g. maximum and average) have seen some success in tracking the changes in vegetation over time, using such scalar representations of the images may fail to capture information on the distribution of NDVI. Therefore, we look for a more comprehensive measure (e.g. a FTS) to describe the distribution of the NDVI values present in each image. Figure 4 gives two examples of NDVI images and the corresponding densities of NDVI estimated from each image.

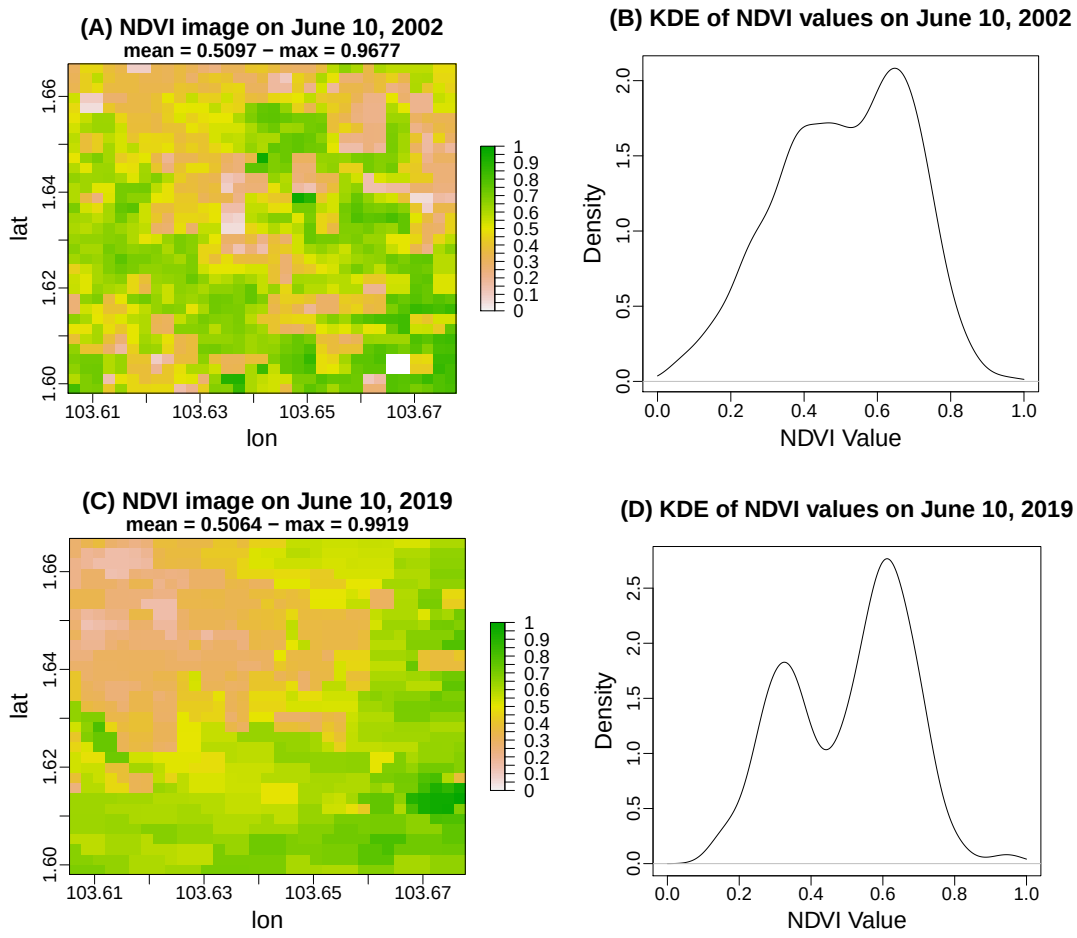

Figure 4: (A): NDVI image taken on June 10, 2002; (B): Kernel density estimate (KDE) of NDVI for the image shown in plot (A); (C): NDVI image taken on June 10, 2019; (D): KDE of NDVI for the image shown in plot (C); Images in (A) and (C) are taken of the Jambi province in Indonesia.

In our study, we use 448 NDVI images, taken in 16 days increments, between February 18, 2002 to July 28, 2019. The images have a spatial resolution of 250 meters and are of a square region just outside of the city of Jambi, Indonesia between $103.61°E - 103.68°E$ and $1.60°S - 1.67°S$. Figure 4 shows the respective NDVI images taken on June 10, 2002 and June 10, 2019. Although the respective NDVI images are not similar, the sample means of the NDVI values are very close (differ by only about 0.0032 unit). As shown in Figure 4, the kernel density estimates (KDEs) are much more informative in representing the distribution of the NDVI values in each image. In addition, the KDE's do a better job in capturing the differences between NDVI images as compared to the sample mean-based approaches. We also note that using the maximum value to represent each image can give misleading results (an image might have one pixel value with high NDVI with all other pixel values close to zero).

We follow Silverman's rule of thumb (Silverman, 1986) for bandwidth selection and obtain the KDEs which are then projected onto a function space spanned by a cubic B-spline basis comprised of 22 functions, where the number of basis elements is selected via the GCV criterion. Figure 5(A) shows the projected KDEs onto the B-spline function space. We pass the results (FTS of length 448) as an input to the FSSA algorithm with a lag of $L = 45$ (to capture annual seasonality components present in vegetative processes), and we study the behavior of the NDVI images over almost two decades, using the **Rfssa** package.

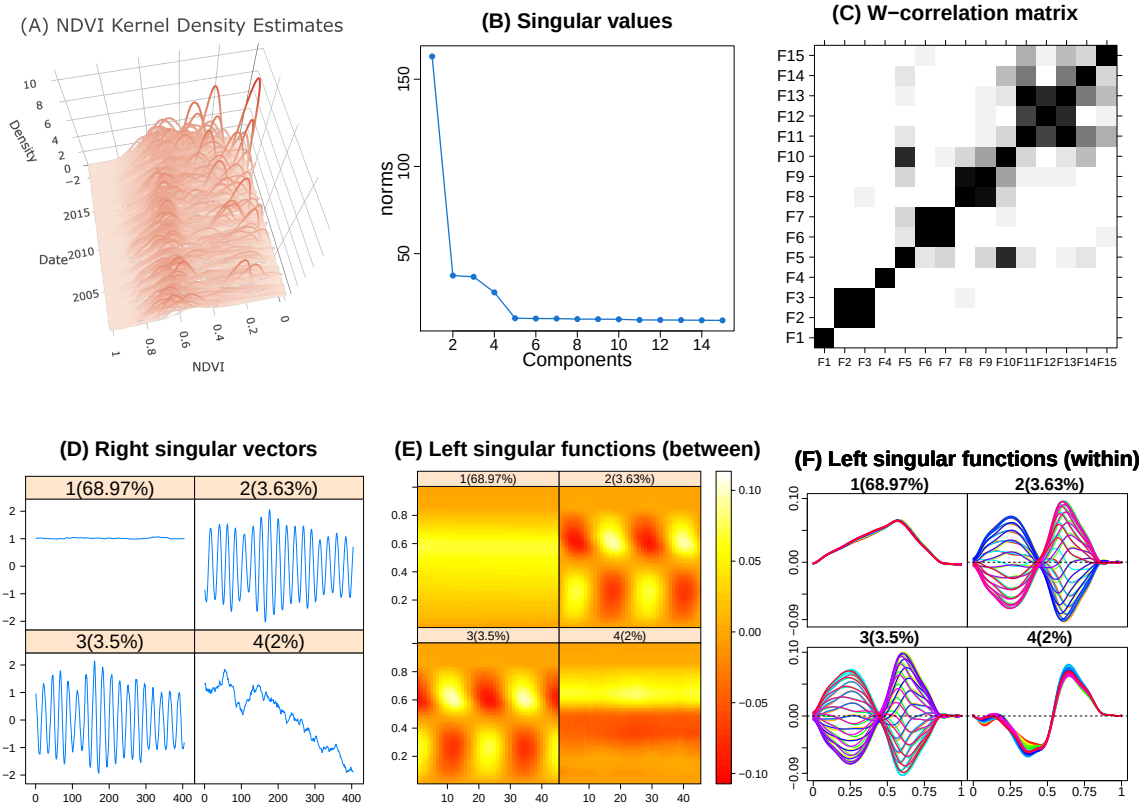

Figure 5: (A): The KDEs of the 448 NDVI images; (B): Plot of the singular values; (C): Plot of the w-correlation matrix; (D): Plot of the right singular vectors; (E): Heatmap of the left singular functions; (F): Plot of the left singular functions.

According to the subplots in Figure 5(B-C: the singular values and w-correlation plots), a suitable partition would be grouping the first and fourth components separately, plus the second and third components jointly ($G = \{1, 2–3, 4\}$). The remaining subplots in Figures 5(D-F: the right singular vectors and left singular functions) indicate the first component captures mean behavior, the second and third capture annual behavior, and the fourth captures a decreasing trend which we discuss in

the following.

An important goal of analyzing the NDVI data is to investigate the existence of a temporal trend in the NDVI images. FSSA can distinguish and separate the overall mean structure (top-left subfigures in Figures 5(D-F)) and the trend pattern (bottom-right subfigures in Figures 5(D-F)) in two different components (the first and fourth components). The trend component is causing an interesting change-point behavior after almost a decade (bottom-right subfigures in Figure 5(D-F). This is due to the fact that the fourth left singular function initially has positive function values corresponding to higher NDVI values (about 0.65) in the domain. However, after about a decade (about the 200^{th} component in the fourth right singular vector), the fourth right singular vector weights the fourth left singular function negatively causing positive function outputs to correspond with lower NDVI values (about 0.4). In addition, we report that the SSA algorithm applied to the time series of sample means of the NDVI images with a lag of $L = 45$ cannot separate this trend structure from the overall mean pattern. We find that both of these modes of variation get combined into one component in the SSA analysis.

Next, we build the reconstruction of the NDVI densities using the grouping, G, suggested by the FSSA exploratory plots, as shown in Figure 6. Figure 6(A) shows the reconstructed overall mean function that does not change over time. Figure 6(B) provides the annual periodic behavior, and we observe the change-point behavior after a decade (change point happens around the year 2012) in Figure 6(C). The last subfigure, Figure 6(D), presents the sum of the trend and mean components. This analysis shows that FSSA was able to separate out the mean from the trend component in the NDVI data while the SSA method applied to the time series of sample means of the NDVI images was not able to separate out these modes of variation from one another.

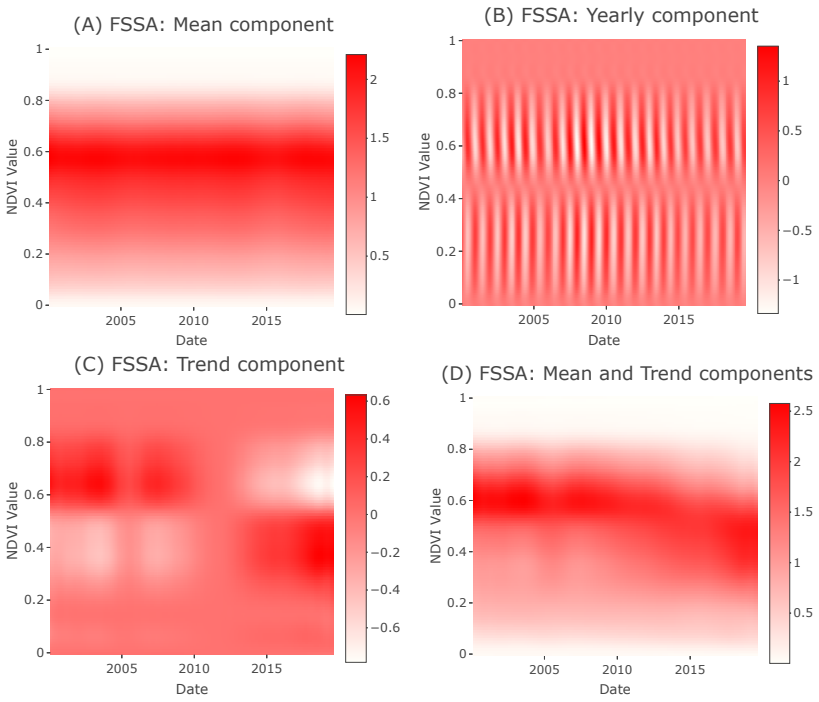

Figure 6: (A): the overall mean component (first group); (B): annual harmonic components (second group); (C): trend component (third group); and (D): sum of the trend and overall mean components (first and third groups).

It would be of interest to confirm the properties of the reconstructed groups using some rigorous statistical procedures. In order to do this, we provide the multivariate trace periodicity test of Hörmann et al. (2018) to test for the annual seasonal effect of FTS. We also perform a bootstrap procedure to test existence of a trend over the time series of the mean of the coefficients associated to the B-spline basis (Grosjean and Ibanez, 2018). We obtain the results of these two tests (periodicity and trend) on four FTS {original signal $y_t(s), R_1, R_2, R_3$}, where R_i represents residual curves obtained by removing the reconstructed FTS by the first i groups, from the original signal $y_t(s)$. Table 2 provides the p-values of the tests for the periodicity and the trend and we find that the annual pattern is present in $y_t(s)$ and R_1 which contain the seasonal components (p-value=0). Also the trend test is significant for all of the FTS, except R_3 that does not contain the third group (fourth component) indicating the presence of trend.

FTS	periodicity test	trend test
$y_t(s)$	0.00	0.02
R_1	0.00	0.03
R_2	0.45	0.03
R_3	0.30	0.07

Table 2: P-values of the multivariate trace periodicity test of Hörmann et al. (2018) and the bootstrap trend test (Grosjean and Ibanez, 2018) on four FTS: $\{y_t(s), R_1, R_2, R_3\}$.

While the FSSA procedure was able to separate out the year long seasonal pattern, it was also able to detect the less obvious trend component present in the NDVI data; which indicates a loss of vegetation over the previous 20 years. It was determined that between 2001 and 2015, grassland and shrubs in the Jambi region accounted for a large amount of the vegetation lost due to controlled burning for human use (Prasetyo et al., 2016). The region specified in images that we study here is primarily upland agriculture and grass and it appears to have been a hot spot for controlled fires (Prasetyo et al., 2016).

2.5 Discussion

In this chapter, we constructed the FSSA procedure by incorporating ideas from SSA and FDA. The contribution of the proposed model is to provide practitioners with some tools to utilize the advantages of SSA in analysis of FTS. Accordingly, researchers can analyze sequences of functions (e.g., time series, longitudinal, or spatial data) via FSSA. Alternatively one may approach the problem using MSSA, given that the data points are measured over a sparse regular grid over time.

As for the ease of use, a user-friendly R implementation of FSSA is developed in the **Rfssa** package. Furthermore, a shiny web application is also included in the package and it is available at `https://fssa.shinyapps.io/fssa/` for reproducing the results of this chapter or analyzing any other FTS.

CHAPTER 3: FUNCTIONAL TIME SERIES FORECASTING: FUNCTIONAL SINGULAR SPECTRUM ANALYSIS APPROACHES

In this chapter we propose two nonparametric methods used in the forecasting of FTS data, namely FSSA recurrent forecasting and FSSA vector forecasting. Both algorithms utilize the results of FSSA and past observations in order to predict future data points where FSSA recurrent forecasting predicts one function vector at a time and the FSSA vector forecasting algorithm makes predictions using length L function vectors. We compare our forecasting algorithms to a gold standard method used in the prediction of FTS by way of simulation and a real data study and we find our techniques perform better for periodic stochastic processes.

3.1 Introduction

An important problem often confronted by researchers is prediction of stochastic processes. Golyandina et al. (2001) expanded on the results of the SSA algorithm to deliver two commonly used nonparametric techniques in forecasting, (a) SSA recurrent forecasting, and (b) SSA vector forecasting. Since the aforementioned SSA techniques have seen success in forecasting time series data, we have the goal of expanding the routines to the FDA world for forecasting FTS.

One of the first approaches to FTS forecasting is given in Hyndman and Ullah (2007) who found success in predicting mortality rate data. The method of Hyndman and Ullah (2007) has been extended so that more recent FTS observations play a greater role in forecasts (Hyndman and Shang, 2009). In addition, extensions have been made so that the method is robust in the presence of outliers (Beyaztas and Shang, 2019; Shang, 2019). The approach of Hyndman and Ullah (2007) has also inspired a functional extension to the ARMAX model allowing for estimation of moving average terms (González et al., 2018). In addition, the methodology of Hyndman and Ullah (2007) and its variants have seen success in applications other than just mortality rate data, see Shang (2013) and Wagner-Muns et al. (2018).

The approach of Hyndman and Ullah (2007) and its extensions consists of two steps. In the first step, they use functional principal component analysis, or its variants, to reduce the dimensionality of the functional data and project the curves onto the functional principal component basis. In the second step, they perform forecasting of the basis coefficients using various models such as ARIMA. The details of the technique of Hyndman and Ullah (2007) are given in Section B.2 of the Appendix. Since the functional prinicpal component analysis-based approaches to forcasting do not consider time-

dependency in the dimension reduction step, the resulting functional principal component analysis basis elements may miss important information that could be captured by the basis found using FSSA. In this chapter we develop two forecasting algorithms based on FSSA that incorporates the time-dependency into the decomposition of FTS. Furthermore, the proposed algorithms do not need the stationary assumption.

In order to depict our approach and its utility, consider the following motivating example involving the intraday bank call center data from Section 2.1 that is shown again in Figure 7.

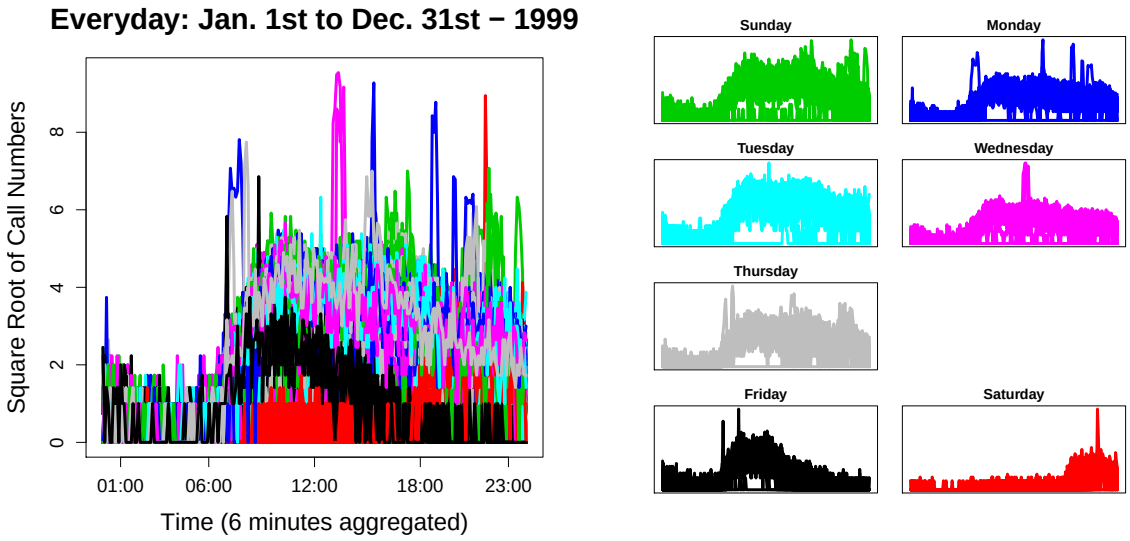

Figure 7: (left): The plot of 365 functional intraday call center curves (one for each day of the year 1999) where the output of each function is the square root of call numbers; (right): The plots of intraday call center curves partitioned by which day of the week each function curve was observed.

Here we see that there exists a strong weekly periodicity in the data between workdays (Sunday through Thursday) and weekends (Friday and Saturday) where the type of functional observation we observe depends heavily on which day of the week the curve was recorded. We wish to determine how well our methods perform in predicting future observations in the call center data as compared with the technique of Hyndman and Ullah (2007). We partition the $N = 365$ curves into a training set of size $M = 308$ (curves observed starting January 1, 1999 and ending November 4, 1999) and a testing set of the remaining 57 (curves observed starting November 5, 1999 and ending December 31, 1999) functions. We compare our proposed methodologies of FSSA recurrent forecasting and FSSA vector forecasting to the approach of Hyndman and Ullah (2007) in a rolling forecast fashion to obtain Figure 8.

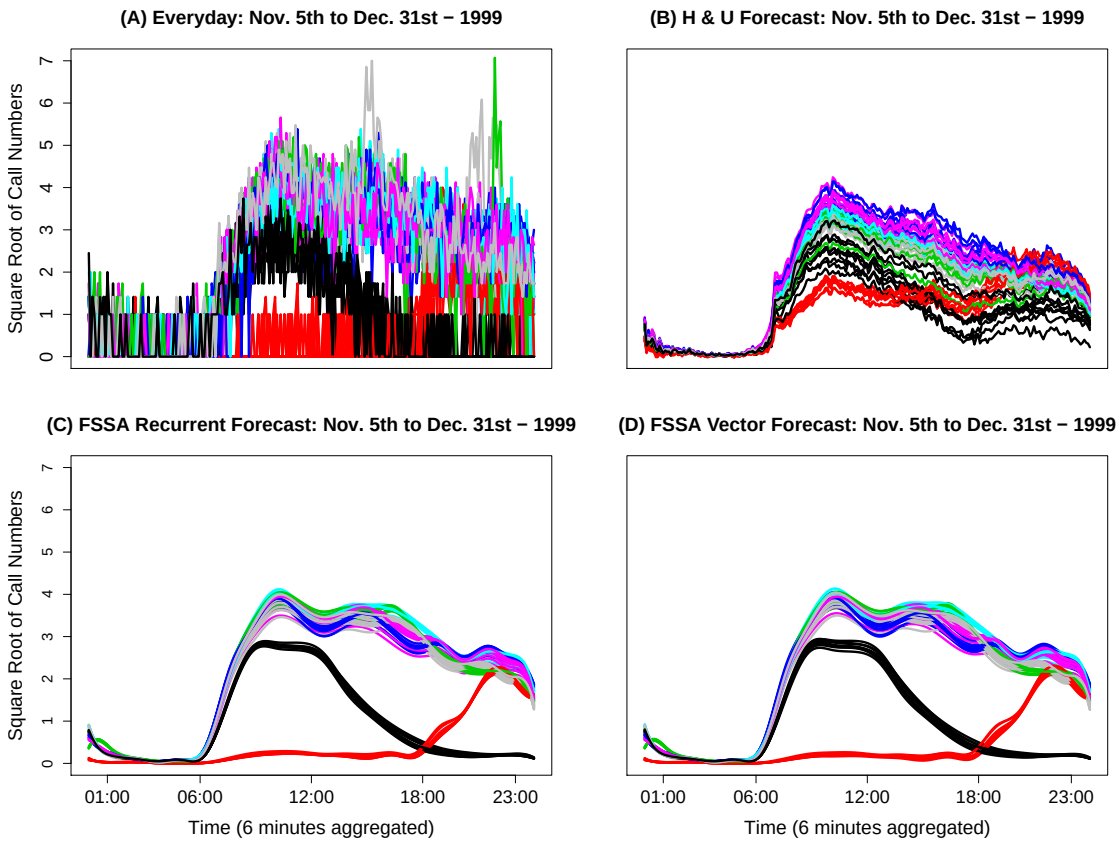

Figure 8: (A): The observed testing set; (B): The predictions made using the method of Hyndman and Ullah (2007); (C): The FSSA recurrent forecasting predictions; (D): The FSSA vector forecasting predictions.

We see that the popular method of Hyndman and Ullah (2007) struggles in predicting the periodic behavior of the call center data especially when trying to differentiate workdays from weekends, while our methods of FSSA recurrent forecasting and FSSA vector forecasting capture this periodicity, and reflect that in the prediction.

The rest of the chapter is organized as follows. In Section 3.2, we develop the theoretical foundations of FSSA recurrent and FSSA vector forecasting. Then in Section 3.3 we give the techniques needed to implement the FSSA recurrent and FSSA vector forecasting approaches. Section 3.4 gives a simulation study and a real data study showing how our methods outperform a functional seasonal naive method and the popular technique of Hyndman and Ullah (2007) in forecasting periodic FTS. Finally, we end with a discussion on results in Section 3.5.

3.2 Functional Singular Spectrum Analysis Forecasting

Similar to Subsection 1.3.2, we are concerned with forecasting the assumed deterministic nature of some FTS, $\mathbf{y}_N$. As such, we apply the first two steps of the FSSA routine to $\mathbf{y}_N$, as shown in Section 2.2, and now we develop the grouping and reconstruction stages in FSSA for the purpose of forecasting.

FSSA Grouping for Forecasting

Consider $\mathfrak{S}$ to be a subset of indices, $\{1, \ldots, r\}$, and let $\boldsymbol{\mathcal{X}}_{\mathfrak{s}} : \mathbb{R}^K \to \mathbb{H}^{L \times K}$ be an operator defined by $\boldsymbol{\mathcal{X}}_{\mathfrak{s}} = \sum_{i \in \mathfrak{S}} \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i$ where the definitions of $\mathbb{H}^{L \times K}$, $\sqrt{\lambda_i}$, $\boldsymbol{\psi}_i$, $\mathbf{v}_i$, and $\mathbf{v}_i \otimes \boldsymbol{\psi}_i$ can be found in Section 2.2. We also draw the definitions of inner products equipped to $\mathbb{H}$ and $\mathbb{H}^L$ directly from Section 2.2. This allows us to write the FSSA trajectory operator, $\boldsymbol{\mathcal{X}}$, as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{\mathfrak{s}} + \boldsymbol{\mathcal{X}}_{\mathfrak{n}},$$

where the operator, $\boldsymbol{\mathcal{X}}_{\mathfrak{n}} = \sum_{i \in \mathfrak{S}^c} \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i$, represents the residual component, and $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ represents the signal component. Similar to equation (15), the range of the operator, $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$, can be written using linear combinations of length L function vectors, denoted with $\{\boldsymbol{x}_j^{\mathfrak{s}}\}_{j=1}^K$, which are leveraged in the FSSA vector forecasting method. In the next step, we extract a FTS from $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ that captures the assumed deterministic signal present in $\mathbf{y}_N$ where the extracted FTS is used in the FSSA recurrent forecasting algorithm.

FSSA Reconstruction for Forecasting

Now we form the orthogonal projection of $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ onto the subspace of Hankel operators that map from $\mathbb{R}^K$ to $\mathbb{H}^L$, which we denote with $\widetilde{\boldsymbol{\mathcal{X}}} \in \mathbb{H}_H^{L \times K}$, where $\mathbb{H}_H^{L \times K}$ is defined in Section 2.2. We find that $\widetilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{\Pi}_H \boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ where the operator, $\boldsymbol{\Pi}_H : \mathbb{H}^{L \times K} \to \mathbb{H}_H^{L \times K}$, is defined in the reconstruction step of FSSA. Now we find that the reconstructed FTS that is representative of the signal is expressed by $\tilde{\mathbf{y}}_N = \boldsymbol{\mathcal{T}}^{-1} \widetilde{\boldsymbol{\mathcal{X}}} = (\tilde{y}_1, \ldots, \tilde{y}_N)$ where $\boldsymbol{\mathcal{T}} : \mathbb{H}^N \to \mathbb{H}_H^{L \times K}$ is the one-to-one and onto embedding operator defined in step one of FSSA.

As an immediate result, we have that the residual elements of the signal, $\mathbf{y}_N$, are captured in the FTS, $\mathbf{y}_N - \tilde{\mathbf{y}}_N$. It is assumed that the residual FTS captures noisy modes of variation and we typically do not include these components in the SSA-based forecasts, see Golyandina et al. (2015, 2001); Golyandina and Zhigljavsky (2013); Hassani and Mahmoudvand (2013) for more information.

3.2.1 FSSA Recurrent Forecasting and FSSA Vector Forecasting

The following contains some notations that are leveraged throughout the rest of this Section and Section 3.3. For each length L function vector, $\boldsymbol{x} \in \mathbb{H}^L$, denote by $\boldsymbol{x}^{\nabla} \in \mathbb{H}^{L-1}$ and $\boldsymbol{x}^{\Delta} \in \mathbb{H}^{L-1}$ as the length $L-1$ function vectors consisting of the first and the last (respectively) $L-1$ components of $\boldsymbol{x}$. We choose a positive integer $k < r$, and we define the function vector spaces $\mathbb{L} = \text{span}\{\boldsymbol{\psi}_i\}_{i=1}^k$ and $\mathbb{L}^{\nabla} = \text{span}\{\boldsymbol{\psi}_i^{\nabla}\}_{i=1}^k$. Moreover, let $\pi_i \in \mathbb{H}$ be a function vector that is defined as the last component of the i^{th} left singular function, $\boldsymbol{\psi}_i \in \mathbb{H}^L$. With this notation, we may continue into the theory of forecasting.

Here, we offer an important theorem and a corollary that is leveraged in the FSSA recurrent and FSSA vector forecasting algorithms.

Theorem 3.1. *Define the set of length L function vectors, $\mathbb{E} = \{(0, \ldots, 0, x) \in \mathbb{H}^L | x \in \mathbb{H}$ and $x \neq 0\}$, and define the operator $\boldsymbol{\mathcal{V}} : \mathbb{H} \to \mathbb{H}$ such that $\boldsymbol{\mathcal{V}} = \sum_{i=1}^k \pi_i \otimes \pi_i$. If the intersection of $\mathbb{E}$ and the set that contains all elements of $\mathbb{L}$, where such an intersection is denoted by $\mathbb{E} \cap \mathbb{L}$, is equal to the empty set, then we have*

$$\|\boldsymbol{\mathcal{V}}\| = \sup_{\|x\|_{\mathbb{H}}=1} \|\boldsymbol{\mathcal{V}}(x)\|_{\mathbb{H}} < 1,$$

where $\|\cdot\|$ is the operator norm for all operators that map from $\mathbb{H}$ to $\mathbb{H}$.

Proof. Fix $\boldsymbol{e}_x = (0, \ldots, 0, x)$ to be a length L function vector in $\mathbb{E}$. We define $\boldsymbol{\Pi}_{\mathbb{L}} : \mathbb{H}^L \to \mathbb{L}$ to be the orthogonal projector onto $\mathbb{L}$ defined by $\boldsymbol{\Pi}_{\mathbb{L}} = \sum_{j=1}^k \psi_j \otimes \psi_j$. Using $\boldsymbol{\Pi}_{\mathbb{L}}$, we may express the orthogonal projection of $\boldsymbol{e}_x$ onto $\mathbb{L}$ as the following length L function vector

$$\boldsymbol{\Pi}_{\mathbb{L}}(\boldsymbol{e}_x) = \boldsymbol{p} = \sum_{j=1}^k \langle \boldsymbol{e}_x, \boldsymbol{\psi}_j \rangle_{\mathbb{H}^L} \boldsymbol{\psi}_j.$$

Notice that $\langle \boldsymbol{e}_x, \boldsymbol{\psi}_j \rangle_{\mathbb{H}^L} = \langle x, \pi_j \rangle_{\mathbb{H}}$ for $j = 1, \ldots, k$, and as such we find that

$$\boldsymbol{p} = \begin{pmatrix} \sum_{j=1}^k \langle x, \pi_j \rangle_{\mathbb{H}} \psi_{1,j} \\ \vdots \\ \sum_{j=1}^k \langle x, \pi_j \rangle_{\mathbb{H}} \psi_{L-1,j} \\ \sum_{j=1}^k \langle x, \pi_j \rangle_{\mathbb{H}} \pi_j \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^k \langle x, \pi_j \rangle_{\mathbb{H}} \psi_{1,j} \\ \vdots \\ \sum_{j=1}^k \langle x, \pi_j \rangle_{\mathbb{H}} \psi_{L-1,j} \\ \boldsymbol{\mathcal{V}}(x) \end{pmatrix}.$$

We define the i^{th} component of $\boldsymbol{p}$ to be a function vector in $\mathbb{H}$ given by, $p_i = \sum_{j=1}^{k} \langle x, \pi_j \rangle_{\mathbb{H}} \psi_{i,j}$ for $i = 1, \ldots, L-1$, and we find that

$$\|\boldsymbol{p}\|_{\mathbb{H}^L}^2 = \sum_{i=1}^{L-1} \|p_i\|_{\mathbb{H}}^2 + \|\boldsymbol{\mathcal{V}}(x)\|_{\mathbb{H}}^2.$$

Since $\boldsymbol{p}$ is the orthogonal projection of $\boldsymbol{e}_x$ onto $\mathbb{L}$ and it is assumed that $\boldsymbol{e}_x \notin \mathbb{L}$, we then have $\|\boldsymbol{\Pi}_{\mathbb{L}}(\boldsymbol{e}_x)\|_{\mathbb{H}^L}^2 = \|\boldsymbol{p}\|_{\mathbb{H}^L}^2 = \sum_{i=1}^{L-1} \|p_i\|_{\mathbb{H}}^2 + \|\boldsymbol{\mathcal{V}}(x)\|_{\mathbb{H}}^2 < \|\boldsymbol{e}_x\|_{\mathbb{H}^L}^2 = \|x\|_{\mathbb{H}}^2$. Since $\|\boldsymbol{\Pi}_{\mathbb{L}}(\boldsymbol{e}_x)\|_{\mathbb{H}^L}^2 < \|\boldsymbol{e}_x\|_{\mathbb{H}^L}^2$, we immediately have that $\|\boldsymbol{\Pi}_{\mathbb{L}}\|_L^2 < 1$ where $\|\cdot\|_L$ is the operator norm for all operators that map from $\mathbb{H}^L$ to $\mathbb{H}^L$. In addition, we notice that

$$\|\boldsymbol{\mathcal{V}}(x)\|_{\mathbb{H}}^2 \leq \|\boldsymbol{\Pi}_{\mathbb{L}}(\boldsymbol{e}_x)\|_{\mathbb{H}^L}^2 < \|x\|_{\mathbb{H}}^2$$

which implies that $\|\boldsymbol{\mathcal{V}}\|^2 \leq \|\boldsymbol{\Pi}_{\mathbb{L}}\|_L^2 < 1$. This gives the result that $\|\boldsymbol{\mathcal{V}}\| < 1$ and the proof is completed.

□

We immediately obtain the following corollary that explicitly defines an operator used in FSSA recurrent forecasting and FSSA vector forecasting algorithms.

Corollary 3.2. *Given that the conditions of Theorem 3.1 are met, we have that* $\sum_{l=0}^{\infty} \boldsymbol{\mathcal{V}}^l$ *is a convergent Neumann series and that the operator,* $(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} : \mathbb{H} \to \mathbb{H}$*, exists where* $\boldsymbol{\mathcal{I}} : \mathbb{H} \to \mathbb{H}$ *is the identity operator. Furthermore, we obtain the equality*

$$(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} = \sum_{l=0}^{\infty} \boldsymbol{\mathcal{V}}^l.$$

Now we define $\mathbf{g}_{N+M} = (g_1, \ldots, g_N, g_{N+1}, \ldots, g_{N+M}) \in \mathbb{H}^{N+M}$ as a FTS of length $N+M$, where in the following algorithms (FSSA recurrent and FSSA vector forecasting), the first N function vectors (g_1 to g_N), are close to $\tilde{\mathbf{y}}_N$, and the main goal is to predict the last M terms (g_{N+1} to g_M).

FSSA Recurrent Forecasting Algorithm

The SSA recurrent forecasting approach of Golyandina et al. (2001) leverages the fact that the next forecasted value of a reconstructed time series can be expressed as a linear combination of the previous $L-1$ elements. When we migrate to the FDA realm, we move towards linear combinations

of function vectors that are weighted by operators. With this in mind, we now give the following theorem which we employ in the FSSA recurrent forecasting and FSSA vector forecasting algorithms.

Theorem 3.3. *Given that the conditions of Theorem 3.1 are met, for any length L function vector, $\boldsymbol{y} = (y_1, \ldots, y_L) \in \mathbb{L}$, the last component, $y_L \in \mathbb{H}$, can be expressed by the following.*

$$y_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j y_j,$$

where $\boldsymbol{\mathcal{A}}_j : \mathbb{H} \to \mathbb{H}$ is an operator given by $\boldsymbol{\mathcal{A}}_j = \sum_{n=1}^{k} \psi_{j,n} \otimes (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \pi_n$, and $\psi_{j,n} \in \mathbb{H}$ is the j^{th} component of the n^{th} left singular function.

Proof. Recall that the length $L-1$ function vectors, $\{\boldsymbol{\psi}_n^{\nabla}\}_{n=1}^{k}$, linearly span the function vector space $\mathbb{L}^{\nabla}$, then for each length $L-1$ function vector, $\boldsymbol{v} \in \mathbb{L}^{\nabla}$, there exists a unique collection of coefficients, $\{h_n\}_{n=1}^{k}$, such that

$$\boldsymbol{v} = \sum_{n=1}^{k} h_n \boldsymbol{\psi}_n^{\nabla}.$$

Also recall that the k left singular functions, $\{\boldsymbol{\psi}_n\}_{n=1}^{k}$, linearly span the function space $\mathbb{L}$ and as such, any length L function vector, $\boldsymbol{y} \in \mathbb{L}$, can be expressed as

$$\boldsymbol{y} = \sum_{n=1}^{k} h_n \boldsymbol{\psi}_n.$$

From here, we see that $\boldsymbol{y}^{\nabla} = \sum_{n=1}^{k} h_n \boldsymbol{\psi}_n^{\nabla} = \boldsymbol{v}$. As for the uniqueness of $\boldsymbol{y}$, let $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ be length L function vectors in $\mathbb{L}$ and let $\boldsymbol{y}_1^{\nabla} = \boldsymbol{y}_2^{\nabla} = \boldsymbol{v}$, then $\boldsymbol{y}_1 - \boldsymbol{y}_2 \in \mathbb{L}$ and is proportional to some length L function vector, $\boldsymbol{e}_x \in \mathbb{E}$, which is a contradiction and implies that $\boldsymbol{y}_1 = \boldsymbol{y}_2$. Thus, we see that $\boldsymbol{y}$ must be unique.

Now we show that for some length L function vector, $\boldsymbol{y} = (y_1, \ldots, y_L)^{\top} \in \mathbb{L}$, the last component, $y_L \in \mathbb{H}$, is a linear combination of the previous $L-1$ components. Let $\boldsymbol{v} = (y_1, \ldots, y_{L-1}) \in \mathbb{L}^{\nabla}$ be a length $L-1$ function vector and notice that

$$(y_1, \ldots, y_{L-1}, 0) + (0, \ldots, 0, y_L) = \sum_{n=1}^{k} h_n \boldsymbol{\psi}_n. \tag{32}$$

Taking the inner product equipped to $\mathbb{H}^L$ of the left and right-hand sides of equation (32) with each $\boldsymbol{\psi}_n$ gives us

$$h_n = \left\langle \boldsymbol{v}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} + \langle y_L, \pi_n \rangle_{\mathbb{H}}, \quad n = 1, \ldots, k. \tag{33}$$

Notice that we may express y_L as

$$y_L = \sum_{n=1}^{k} h_n \pi_n \tag{34}$$

and now we substitute in the right-hand side of equation (33) for h_n in equation (34) to obtain

$$y_L = \sum_{n=1}^{k} \left\langle \boldsymbol{\psi}_n^{\nabla}, \boldsymbol{v} \right\rangle_{\mathbb{H}^{L-1}} \pi_n + \boldsymbol{\mathcal{V}}(y_L).$$

We subtract the function vector, $\boldsymbol{\mathcal{V}}(y_L) \in \mathbb{H}$, to the left-hand side of the equation and factor out y_L on the right. Since, we assume that $\mathbb{E} \cap \mathbb{L} = \emptyset$ such that the operator, $(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1}$ exists, we find that

$$y_L = \sum_{n=1}^{k} \left\langle \boldsymbol{\psi}_n^{\nabla}, \boldsymbol{v} \right\rangle_{\mathbb{H}^{L-1}} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \pi_n.$$

Recall that $\left\langle \boldsymbol{\psi}_n^{\nabla}, \boldsymbol{v} \right\rangle_{\mathbb{H}^{L-1}} = \sum_{j=1}^{L-1} \langle \psi_{j,n}, y_j \rangle_{\mathbb{H}}$ and thus

$$\begin{aligned} y_L = \sum_{n=1}^{k} \sum_{j=1}^{L-1} \langle \psi_{j,n}, y_j \rangle_{\mathbb{H}} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \pi_n &= \sum_{j=1}^{L-1} \sum_{n=1}^{k} \psi_{j,n} \otimes (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \pi_n (y_j) \\ &= \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j y_j, \end{aligned}$$

which shows that the last component of any length L function vector in $\mathbb{L}$ is a linear combination of the last $L-1$ components.

□

We use the result of Theorem 3.3 to obtain the following algorithm.

Algorithm A - FSSA Recurrent Forecasting:

$$g_i = \begin{cases} \tilde{y}_i, & i = 1, \ldots, N \\ \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j g_{i+j-L}, & i = N+1, \ldots, N+M \end{cases}. \tag{35}$$

For brevity, we call this method the *FSSA R-forecasting* algorithm.

FSSA Vector Forecasting Algorithm

The FSSA R-forecasting algorithm can be extended into an method known as FSSA vector forecasting that allows us to perform prediction using length L function vectors. We define the operator, $\boldsymbol{\mathcal{P}}^{\nabla} : \mathbb{R}^k \to \mathbb{H}^{L-1}$ such that for any vector $\mathbf{a} \in \mathbb{R}^k$, we have

$$\boldsymbol{\mathcal{P}}^{\nabla}(\mathbf{a}) = \sum_{n=1}^{k} a_n \boldsymbol{\psi}_n^{\nabla}.$$

It is easy to see the adjoint operator can be obtained as $\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^* : \mathbb{H}^{L-1} \to \mathbb{R}^k$, such that for any length $L-1$ function vector, $\boldsymbol{v} \in \mathbb{H}^{L-1}$, we have

$$\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^*(\boldsymbol{v}) = \left[\left\langle \boldsymbol{v}, \boldsymbol{\psi}_1^{\nabla} \right\rangle_{\mathbb{H}^{L-1}}, \ldots, \left\langle \boldsymbol{v}, \boldsymbol{\psi}_k^{\nabla} \right\rangle_{\mathbb{H}^{L-1}}\right]^{\top}.$$

For a given length L function vector, $\boldsymbol{x} \in \mathbb{L}$, the goal of vector forecasting is to find a linear operator, $\boldsymbol{\mathcal{Q}} : \mathbb{L} \to \mathbb{L}$, such that for some length L function vector, $\boldsymbol{y} \in \mathbb{L}$, given by $\boldsymbol{y} = \boldsymbol{\mathcal{Q}}\boldsymbol{x}$, the distance between the length $L-1$ function vectors, $\boldsymbol{x}^{\Delta} \in \mathbb{H}^{L-1}$ and $\boldsymbol{y}^{\nabla} \in \mathbb{L}^{\nabla}$, is minimal. The following Proposition defines an operator that used to obtain such a $\boldsymbol{y}^{\nabla}$.

Proposition 3.4. *Given that the conditions of Theorem 3.1 are met, the operator* $\boldsymbol{\Pi}$*, defined as*

$$\boldsymbol{\Pi} = \boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^* \boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1} \left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^*$$

is an orthogonal projection from $\mathbb{H}^{L-1}$ *onto* $\mathbb{L}^{\nabla}$*.*

Proof. To show that $\boldsymbol{\Pi}$ is an orthogonal projection onto $\mathbb{L}^{\nabla}$, we need to show that it is idempotent and self-adjoint. We begin by showing the idempotent property of $\boldsymbol{\Pi}$. First notice that $\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^* \boldsymbol{\mathcal{P}}^{\nabla}$ is an $k \times k$ matrix that is invertible. We let $\boldsymbol{v} \in \mathbb{H}^{L-1}$ be a length $L-1$ function vector and we find that

$$\begin{aligned}\boldsymbol{\Pi}\left(\boldsymbol{\Pi}\left(\boldsymbol{v}\right)\right) &= \boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\left(\boldsymbol{v}\right)\\ &= \boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\left(\boldsymbol{v}\right) = \boldsymbol{\Pi}\left(\boldsymbol{v}\right)\end{aligned}$$

therefore $\boldsymbol{\Pi}$ is idempotent and is a projector onto $\mathbb{L}^{\nabla}$. For the orthogonality, notice that

$$\begin{aligned}\boldsymbol{\Pi}^{*} &= \left(\boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\right)^{*}\\ &= \boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*} = \boldsymbol{\Pi}.\end{aligned}$$

Thus $\boldsymbol{\Pi}$ is self-adjoint and an orthogonal projection onto $\mathbb{L}^{\nabla}$.

□

As an immediate result of Proposition 3.4, we have $\boldsymbol{y}^{\nabla} = \boldsymbol{\Pi}\boldsymbol{x}^{\Delta}$. Now using Theorem 3.3, we obtain the last component of $\boldsymbol{y}$ (i.e., $y_L = \sum_{j=1}^{L-1}\boldsymbol{\mathcal{A}}_j y_j^{\nabla}$). The following Proposition further simplifies the recurrence relation used to find y_L.

Proposition 3.5. *Given that the conditions of Theorem 3.1 are met, we have*

$$y_L = \sum_{j=1}^{L-1}\boldsymbol{\mathcal{A}}_j x_j^{\Delta}.$$

Proof. To begin the proof, we determine the form of $\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}$. Notice that $\mathbf{I}_k = \left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla} + \mathbf{B}$ where $\mathbf{I}_k$ is the $k \times k$ identity matrix and $\mathbf{B}$ is a matrix of the same dimensions given by $\mathbf{B} = \left[\langle \pi_i, \pi_j\rangle_{\mathbb{H}}\right]_{i,j=1}^{k}$. To this end, $\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1} = \left(\mathbf{I}_k - \mathbf{B}\right)^{-1}$. We define the $k \times k$ matrix

$$\mathbf{C} = \left[\left\langle \pi_i, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}}\right]_{i,j=1}^{k},$$

and now multiply

$$\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\left(\mathbf{I}_k + \mathbf{C}\right) = \left(\mathbf{I}_k - \mathbf{B}\right)\left(\mathbf{I}_k + \mathbf{C}\right) = \mathbf{I}_k + \mathbf{C} - \mathbf{B}\mathbf{C} - \mathbf{B}$$

which is an equality that we use to show the form of $\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}$. Notice that

$$
\begin{aligned}
[\mathbf{C}-\mathbf{BC}]_{i,j} &= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} - \sum_{n=1}^{k} \left\langle \pi_i, \pi_n \right\rangle_{\mathbb{H}} \left\langle \pi_n, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} - \left\langle \pi_i, \sum_{n=1}^{k} \left\langle \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right), \pi_n \right\rangle_{\mathbb{H}} \pi_n \right\rangle_{\mathbb{H}}
\end{aligned}
$$

We assume that $\mathbb{E} \cap \mathbb{L} = \emptyset$ and as a result, we have that $\left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1} = \sum_{l=0}^{\infty} \boldsymbol{\mathcal{V}}^l$ is a linear operator and we obtain the following

$$
\begin{aligned}
&\left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} - \left\langle \pi_i, \sum_{n=1}^{k} \left\langle \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right), \pi_n \right\rangle_{\mathbb{H}} \pi_n \right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} - \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\sum_{n=1}^{k} \left\langle \pi_j, \pi_n \right\rangle_{\mathbb{H}} \pi_n\right)\right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right)\right\rangle_{\mathbb{H}} - \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\boldsymbol{\mathcal{V}}\left(\pi_j\right)\right)\right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_j\right) - \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\boldsymbol{\mathcal{V}}\left(\pi_j\right)\right)\right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}\left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)\left(\pi_j\right)\right\rangle_{\mathbb{H}} \\
&= \left\langle \pi_i, \pi_j \right\rangle_{\mathbb{H}} = \mathbf{B}_{i,j}.
\end{aligned}
$$

Thus, we have $\mathbf{I}_k + \mathbf{C} - \mathbf{BC} - \mathbf{B} = \mathbf{I}_k + \mathbf{B} - \mathbf{B} = \mathbf{I}_k$ which shows that $\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^* \boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1} = \mathbf{I}_k + \mathbf{C}$. Using the fact that $\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^* \boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1} = \mathbf{I}_k + \mathbf{C}$, Theorem 3.3, and the definition of $\boldsymbol{\Pi}$, notice that

$$
y_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j y_j^{\nabla} = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \left(\boldsymbol{\mathcal{P}}^{\nabla} \left(\mathbf{I}_k + \mathbf{C}\right) \begin{bmatrix} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_1^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\ \vdots \\ \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_k^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \end{bmatrix} \right)_j = u + v
$$

where

$$u = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \left(\sum_{n=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \psi_{j,n} \right)$$

$$v = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \left(\boldsymbol{\mathcal{P}}^{\nabla} \begin{bmatrix} \sum_{n=1}^{k} \left\langle \pi_1, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) \right\rangle_{\mathbb{H}} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\ \vdots \\ \sum_{n=1}^{k} \left\langle \pi_k, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) \right\rangle_{\mathbb{H}} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \end{bmatrix} \right)_j$$

are function vectors in $\mathbb{H}$. Now notice that

$$\begin{aligned}
u &= \sum_{j=1}^{L-1} \sum_{n=1}^{k} \psi_{j,n} \otimes \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) \left(\sum_{i=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \psi_{j,i} \right) \\
&= \sum_{n=1}^{k} \sum_{i=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\psi}_i^{\nabla}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) \\
&= \sum_{n=1}^{k} \sum_{i=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left(\mathbf{I}_{i,n} - \langle \pi_i, \pi_n \rangle_{\mathbb{H}}\right) \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) \\
&= \sum_{n=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} (\pi_n) - \sum_{i=1}^{k} \sum_{l_1=1}^{L-1} \left\langle x_{l_1}^{\Delta}, \psi_{l_1,i} \right\rangle_{\mathbb{H}} \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} \boldsymbol{\mathcal{V}} (\pi_i) \\
&= \sum_{l_2=1}^{L-1} \boldsymbol{\mathcal{A}}_{l_2} x_{l_2}^{\Delta} - \boldsymbol{\mathcal{V}} \left(\sum_{l_1=1}^{L-1} \boldsymbol{\mathcal{A}}_{l_1} x_{l_1}^{\Delta} \right).
\end{aligned}$$

In addition, notice that

$$
\begin{aligned}
v &= \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \left(\sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left\langle \pi_{n_1}, (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right\rangle_{\mathbb{H}} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \boldsymbol{\psi}_{n_1}^{\nabla} \right)_j \\
&= \sum_{n_3=1}^{k} \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left\langle \pi_{n_1}, (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right\rangle_{\mathbb{H}} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\psi}_{n_1}^{\nabla}, \boldsymbol{\psi}_{n_3}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_3}) \\
&= \sum_{n_3=1}^{k} \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left\langle \pi_{n_1}, (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right\rangle_{\mathbb{H}} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left(\mathbf{I}_{n_1,n_3} - \langle \pi_{n_1}, \pi_{n_3} \rangle_{\mathbb{H}} \right) (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_3}) \\
&= \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \pi_{n_1}, (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right\rangle_{\mathbb{H}} \left((\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_1}) - (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \boldsymbol{\mathcal{V}} (\pi_{n_1}) \right) \quad . \\
&= \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \pi_{n_1}, (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right\rangle_{\mathbb{H}} \left((\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}) (\pi_{n_1}) \right) \\
&= \boldsymbol{\mathcal{V}} \left(\sum_{n_2=1}^{k} \left\langle \boldsymbol{x}^{\Delta}, \boldsymbol{\psi}_{n_2}^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\pi_{n_2}) \right) \\
&= \boldsymbol{\mathcal{V}} \left(\sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta} \right)
\end{aligned}
$$

As a result, we find that

$$
y_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j y_j^{\nabla} = u + v = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta} - \boldsymbol{\mathcal{V}} \left(\sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta} \right) + \boldsymbol{\mathcal{V}} \left(\sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta} \right) = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta}.
$$

Thus we see that $y_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j y_j^{\nabla} = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta}$.

□

From the results of Proposition 3.4 and Proposition 3.5, we derive the linear operator, $\boldsymbol{\mathcal{Q}}$, as follows:

$$
\boldsymbol{\mathcal{Q}} (\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{\Pi} \left(\boldsymbol{x}^{\Delta} \right) \\ \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j x_j^{\Delta} \end{pmatrix}, \qquad \boldsymbol{x} \in \mathbb{L}. \tag{36}
$$

Using $\boldsymbol{\mathcal{Q}}$, we offer the following algorithm:

Algorithm B - FSSA Vector Forecasting:

1. Define the length L function vectors given by

$$\boldsymbol{w}_j = \begin{cases} \boldsymbol{x}_j^{\mathfrak{S}}, & j = 1, \ldots, K \\ \boldsymbol{\mathcal{Q}}\left(\boldsymbol{w}_{j-1}\right), & j = K+1, \ldots, K+M \end{cases} \tag{37}$$

2. Form the operator $\boldsymbol{\mathcal{W}} \in \mathbb{H}^{L \times (K+M)}$ whose range is linearly spanned by the set $\{\boldsymbol{w}_i\}_{i=1}^{K+M}$.

3. Hankelize $\boldsymbol{\mathcal{W}}$ in order to extract the FTS, $\mathbf{g}_{N+M}$.

4. The functions $g_{N+1}, \ldots, g_{N+M}$ form the M terms of the FSSA vector forecast.

We refer to this method as the *FSSA V-forecasting algorithm* for brevity. In the next Section, we present computer implementation of both the FSSA R-forecasting and FSSA V-forecasting algorithms.

3.3 Computer Implementation Strategy

We start with some notation that is leveraged in the implementation of the FSSA R-forecasting and FSSA V-forecasting algorithms. We assume that $\{\nu_i\}_{i=1}^{d}$ is a linearly independent basis for $\mathbb{H}_d$, which is a d-dimensional subspace of $\mathbb{H}$. For any function vector, $f \in \mathbb{H}_d$, there exists a unique vector, $\mathbf{c}_f = [c_{f,1}, \ldots, c_{f,d}]^\top \in \mathbb{R}^d$, where $f = \sum_{i=1}^{d} c_{f,i}\nu_i$. We refer to $\mathbf{c}_f$ as the corresponding vector of coefficients of f. We also define $\mathbf{G} = \left[\langle \nu_i, \nu_j \rangle_{\mathbb{H}}\right]_{i,j=1}^{d}$ to be the $d \times d$ Gram matrix.

From hereafter in the current section, we consider all discussed operators whose domain or range are infinite dimensional Hilbert spaces to operate on and map to the corresponding d-dimensional subspace. For example, now we consider $\boldsymbol{\mathcal{V}}$ to be an operator that maps from $\mathbb{H}_d$ into $\mathbb{H}_d$. Since the FSSA R-forecasting and FSSA V-forecasting algorithms are both dependent on $\left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1} : \mathbb{H}_d \to \mathbb{H}_d$, we explore one particular implementation of this operator in the following lemma.

Lemma 3.6. *Define* $\mathbf{D} = \mathbf{G}^{\frac{1}{2}}\left[\mathbf{c}_{\pi_1} \cdots \mathbf{c}_{\pi_k}\right]$ *to be a* $d \times k$ *matrix. Given that the conditions of Theorem 3.1 are met, then the following holds:*

$$\left[\left\langle \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right), \nu_i \right\rangle_{\mathbb{H}}\right]_{i=1,\ldots,d}^{n=1,\ldots,k} = \mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right).$$

Proof. First, we notice that

$$\left\langle \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right), \nu_i \right\rangle_{\mathbb{H}} = \sum_{l=0}^{\infty}\left\langle \boldsymbol{\mathcal{V}}^l\left(\pi_n\right), \nu_i \right\rangle_{\mathbb{H}}.$$

As such, we use induction to show the desired result of $\left[\left\langle \boldsymbol{\mathcal{V}}^l\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}}\right]_{i=1,\ldots,d}^{n=1,\ldots,k} = \mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\mathbf{D}^\top\mathbf{D}\right)^l$. For our base case, let $l = 1$, then

$$\left\langle \boldsymbol{\mathcal{V}}\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}} = \sum_{n_1=1}^{k}\left\langle \pi_n, \pi_{n_1}\right\rangle_{\mathbb{H}}\left\langle \pi_{n_1}, \nu_i\right\rangle_{\mathbb{H}} = \left[\mathbf{G}^{\frac{1}{2}}\mathbf{D}\mathbf{D}^\top\mathbf{D}\right]_{i,n}.$$

From here, we state our inductive assumption that

$$\left[\left\langle \boldsymbol{\mathcal{V}}^l\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}}\right]_{i=1,\ldots,d}^{n=1,\ldots,k} = \mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\mathbf{D}^\top\mathbf{D}\right)^l.$$

Now notice that

$$\begin{aligned}
\left\langle \boldsymbol{\mathcal{V}}^{l+1}\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}} &= \sum_{n_1=1}^{k}\cdots\sum_{n_{l+1}=1}^{k}\left\langle \pi_n, \pi_{n_{l+1}}\right\rangle_{\mathbb{H}}\left\langle \pi_{n_{l+1}}, \pi_{n_l}\right\rangle_{\mathbb{H}}\cdots\left\langle \pi_{n_2}, \pi_{n_1}\right\rangle_{\mathbb{H}}\left\langle \pi_{n_1}, \nu_i\right\rangle_{\mathbb{H}} \\
&= \left[\mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\mathbf{D}^\top\mathbf{D}\right)^{l+1}\right]_{i,n} = \left[\mathbf{G}^{\frac{1}{2}}\mathbf{D}\mathbf{D}^\top\mathbf{D}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right]_{i,n} \\
&= \left\langle \boldsymbol{\mathcal{V}}\left(\boldsymbol{\mathcal{V}}^l\left(\pi_n\right)\right), \nu_i\right\rangle_{\mathbb{H}}.
\end{aligned}$$

which shows the desired result. As such, we see that

$$\left\langle \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}} = \sum_{l=0}^{\infty}\left\langle \boldsymbol{\mathcal{V}}^l\left(\pi_n\right), \nu_i\right\rangle_{\mathbb{H}} = \left[\mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right)\right]_{i,n}$$

□

Remark: Note that if the conditions of Theorem 3.1 are met, then $\|\boldsymbol{\mathcal{V}}\| < 1$ and in addition, the real-valued sequence, $\left(\|\boldsymbol{\mathcal{V}}^l\|\right)_{l\in\mathbb{N}}$, converges to zero monotonically. As a result, we truncate $\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l$ when $\|\left(\mathbf{D}^\top\mathbf{D}\right)^l\|_{Fr} \approx 0$ for some $l \in \mathbb{N}$ where $\|\cdot\|_{Fr}$ denotes the Frobenius norm of a $k \times k$ matrix. From here we offer the implementation of the FSSA R-forecasting algorithm.

3.3.1 FSSA R-Forecasting Implementation

As according to equation (35), our goal is to find the matrices that implement the effect of each $\boldsymbol{\mathcal{A}}_j : \mathbb{H}_d \to \mathbb{H}_d$ for $j = 1, \ldots, L-1$. The following theorem gives us the form of the desired matrices.

Theorem 3.7. *Given that the conditions of Theorem 3.1 hold, let*

$$\mathbf{A}_j = \mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right)\mathbf{E}_j^\top\mathbf{G}^{\frac{1}{2}}, \qquad \textit{for } j = 1,\ldots,L-1,$$

where $\mathbf{E}_j = \mathbf{G}^{\frac{1}{2}}\left[\mathbf{c}_{\psi_{j,1}}\cdots\mathbf{c}_{\psi_{j,k}}\right]$ *is a* $d\times k$ *matrix. The corresponding coefficients of the function vector,* $\boldsymbol{\mathcal{A}}_j f$*, are given by the* d*-dimensional vector,* $\mathbf{A}_j\mathbf{c}_f$*.*

Proof. Let $f\in\mathbb{H}$ be a function vector and recall that

$$\boldsymbol{\mathcal{A}}_j(f) = \sum_{n=1}^{k}\langle\psi_{j,n},f\rangle_{\mathbb{H}}\left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}(\pi_n).$$

Now using the result of Lemma 3.6

$$\begin{aligned}\left[\mathbf{G}^{\frac{1}{2}}\mathbf{A}_j\mathbf{c}_f\right]_i &= \left[\mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right)\mathbf{E}_j^\top\mathbf{G}^{\frac{1}{2}}\mathbf{c}_f\right]_i\\ &= \left\langle\sum_{n=1}^{k}\langle\psi_{j,n},f\rangle_{\mathbb{H}}\left(\boldsymbol{\mathcal{I}}-\boldsymbol{\mathcal{V}}\right)^{-1}(\pi_n),\nu_i\right\rangle_{\mathbb{H}}\\ &= \langle\boldsymbol{\mathcal{A}}_j(f),\nu_i\rangle_{\mathbb{H}}.\end{aligned}$$

This implies that the matrices that implement the FSSA R-forecasting algorithm have the form of

$$\mathbf{A}_j = \mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right)\mathbf{E}_j^\top\mathbf{G}^{\frac{1}{2}}, \quad j=1,\ldots,L-1.$$

□

The following corollary further expands on the implementation of the FSSA R-forecasting algorithm.

Corollary 3.8. *The* d*-dimensional vectors of coefficients corresponding to the function vectors,* g_i*'s, given in equation* (35) *can be written as*

$$\mathbf{c}_{g_i} = \begin{cases}\mathbf{c}_{\tilde{y}_i}, & i=1,\ldots,N\\ \sum_{j=1}^{L-1}\mathbf{A}_j\mathbf{c}_{g_{i+j-L}}, & i=N+1,\ldots,N+M\end{cases}. \tag{38}$$

3.3.2 FSSA V-Forecasting Implementation

In order to obtain the recipes for the FSSA V-forecasting algorithm, we first leverage theory developed in Section 2.3. For some positive integer, L, we define $\mathbb{H}_d^L$ to be the space created from the Cartesian product of L copies of $\mathbb{H}_d$. Now we define the quotient-remainder sequence

$$j = (q_j - 1)\, L + r_j, \quad 1 \leq r_j \leq L, \quad , 1 \leq q_j \leq d$$

and from here, we define the collection of linearly independent length L function vectors, $\{\boldsymbol{\phi}_j\}_{j=1}^{Ld}$, where $\boldsymbol{\phi}_j \in \mathbb{H}_d^L$ is the zero function in all coordinates except for the r_j^{th} which is ν_{q_j}. We find that the basis, $\{\boldsymbol{\phi}_j\}_{j=1}^{Ld}$, linearly spans $\mathbb{H}_d^L$. As a result, we find that any length L function vector, $\boldsymbol{f} \in \mathbb{H}_d^L$, can be represented via its corresponding coefficient vector $\mathbf{c}_{\boldsymbol{f}} = [c_{\boldsymbol{f},1}, \ldots, c_{\boldsymbol{f},Ld}]^\top \in \mathbb{R}^{Ld}$, where $\boldsymbol{f} = \sum_{i=1}^{Ld} c_{\boldsymbol{f},i}\boldsymbol{\phi}_i$. We denote the updated $Ld \times Ld$ Gram matrix with $\mathbf{H} = \left[\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle_{\mathbb{H}^L}\right]_{i,j=1}^{Ld}$. Now, we define the new truncated quotient-remainder sequence as

$$i = (q_i - 1)\,(L-1) + r_i, \quad 1 \leq r_i \leq L-1, \quad 1 \leq q_i \leq d.$$

This allows us to define the collection of linearly independent length $L-1$ function vectors, $\{\boldsymbol{\phi}_i^\nabla\}_{i=1}^{(L-1)d}$, where $\boldsymbol{\phi}_i^\nabla \in \mathbb{H}_d^{L-1}$ is zero in all coordinates except for the r_i^{th} which is ν_{q_i}, and $\mathbb{H}_d^{L-1}$ is the vector space formed from the Cartesian product of $L-1$ copies of $\mathbb{H}_d$. Similar to the non-truncated case, any length $L-1$ function vector, $\boldsymbol{f}^\nabla$ and $\boldsymbol{f}^\Delta \in \mathbb{H}_d^{L-1}$, can be represented via the corresponding vectors of coefficients given by $\mathbf{c}_{\boldsymbol{f}^\nabla}, \mathbf{c}_{\boldsymbol{f}^\Delta} \in \mathbb{R}^{(L-1)d}$, associated to the basis functions $\{\boldsymbol{\phi}_i^\nabla\}_{i=1}^{(L-1)d}$. Also, we define the respective $(L-1)d \times (L-1)d$ Gram matrix, $\mathbf{H}^\nabla = \left[\left\langle \boldsymbol{\phi}_i^\nabla, \boldsymbol{\phi}_j^\nabla \right\rangle_{\mathbb{H}^{L-1}}\right]_{i,j=1}^{(L-1)d}$.

Now we use these notations to find the matrix that implements $\boldsymbol{\Pi} : \mathbb{H}_d^{L-1} \to \mathbb{H}_d^{L-1}$.

Theorem 3.9. *Let* $\mathbf{F} = \left(\mathbf{H}^\nabla\right)^{\frac{1}{2}} \left[\mathbf{c}_{\boldsymbol{\psi}_1^\nabla} \cdots \mathbf{c}_{\boldsymbol{\psi}_k^\nabla}\right]$ *be a* $(L-1)d \times k$ *matrix. Given that the conditions of Theorem 3.1 are met, we have*

$$\mathbf{P} = \left(\mathbf{F}\mathbf{F}^\top + \mathbf{F}\mathbf{D}^\top\mathbf{D}\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\mathbf{F}^\top\right)\left(\mathbf{H}^\nabla\right)^{\frac{1}{2}},$$

is the $(L-1)d \times (L-1)d$ *matrix that implements* $\boldsymbol{\Pi}$*. In addition, for any length* $L-1$ *function vector,* $\boldsymbol{f} \in \mathbb{H}_d^{(L-1)}$*, the corresponding coefficients of the* $L-1$ *function vector,* $\boldsymbol{\Pi f}$ *are given by the vector* $\mathbf{P}\mathbf{c}_{\boldsymbol{f}} \in \mathbb{R}^{(L-1)d}$*.*

Proof. Notice that

$$\begin{aligned}
\left\langle \mathbf{\Pi}\left(\boldsymbol{\phi}_j^{\nabla}\right), \boldsymbol{\phi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} &= \left\langle \boldsymbol{\mathcal{P}}^{\nabla} \left(\left(\mathbf{I}_k + \mathbf{C}\right) \begin{bmatrix} \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_1^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\ \vdots \\ \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_k^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \end{bmatrix} \right), \boldsymbol{\phi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\
&= \left\langle \boldsymbol{\mathcal{P}}^{\nabla} \left(\begin{bmatrix} \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_1^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\ \vdots \\ \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_k^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \end{bmatrix} \right) + \boldsymbol{\mathcal{P}}^{\nabla} \left(\begin{bmatrix} \sum_{n=1}^{k} \left\langle \pi_1, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right) \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\ \vdots \\ \sum_{n=1}^{k} \left\langle \pi_k, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right) \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_n^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \end{bmatrix} \right), \boldsymbol{\phi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\
&= \sum_{l=1}^{k} \left\langle \boldsymbol{\phi}_j^{\nabla}, \boldsymbol{\psi}_l^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\psi}_l^{\nabla}, \boldsymbol{\phi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} + \sum_{l=1}^{k} \sum_{n=1}^{k} \left\langle \boldsymbol{\phi}_i^{\nabla}, \boldsymbol{\psi}_l^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \left\langle \pi_l, \left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}}\right)^{-1}\left(\pi_n\right) \right\rangle_{\mathbb{H}^{L-1}} \left\langle \boldsymbol{\psi}_n^{\nabla}, \boldsymbol{\phi}_j^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} \\
&= \left[\left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \mathbf{F}\mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} + \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \mathbf{F}\mathbf{D}^{\top}\mathbf{D} \sum_{l=0}^{\infty} \left(\mathbf{D}^{\top}\mathbf{D}\right)^{l} \mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \right]_{i,j} \\
&= \left[\left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \left(\mathbf{F}\mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} + \mathbf{F}\mathbf{D}^{\top}\mathbf{D} \sum_{l=0}^{\infty} \left(\mathbf{D}^{\top}\mathbf{D}\right)^{l} \mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \right) \right]_{i,j}.
\end{aligned}$$

Now notice that for some length $L-1$ function vector, $\boldsymbol{f} \in \mathbb{H}_d^{L-1}$, we have

$$\left\langle \mathbf{\Pi}\left(\boldsymbol{f}\right), \boldsymbol{\phi}_i^{\nabla} \right\rangle_{\mathbb{H}^{L-1}} = \left[\left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \left(\mathbf{F}\mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \mathbf{c}_{\boldsymbol{f}} + \mathbf{F}\mathbf{D}^{\top}\mathbf{D} \sum_{l=0}^{\infty} \left(\mathbf{D}^{\top}\mathbf{D}\right)^{l} \mathbf{F}^{\top} \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}} \mathbf{c}_{\boldsymbol{f}} \right) \right]_{i}.$$

This implies that the $(L-1)d \times (L-1)d$ matrix that implements $\mathbf{\Pi}$ is

$$\mathbf{P} = \left(\mathbf{F}\mathbf{F}^{\top} + \mathbf{F}\mathbf{D}^{\top}\mathbf{D} \sum_{l=0}^{\infty} \left(\mathbf{D}^{\top}\mathbf{D}\right)^{l} \mathbf{F}^{\top} \right) \left(\mathbf{H}^{\nabla}\right)^{\frac{1}{2}}.$$

□

Using the result of Theorem 3.9, we have that the following Theorem gives the full implementation of the operator, $\boldsymbol{\mathcal{Q}}$.

Theorem 3.10. *For any $\boldsymbol{x} \in \mathbb{L}$, let the linear operator $\boldsymbol{\mathcal{Q}}$ to be the one given in* (36)*. The*

corresponding coefficients of $\boldsymbol{\mathcal{Q}x}$ *are given by*

$$\mathbf{c}_{\boldsymbol{\mathcal{Q}}}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{P}\mathbf{c}_{\boldsymbol{x}^{\triangle}} \\ \sum_{j=1}^{L-1} \mathbf{A}_j \mathbf{c}_{x_j^{\triangle}} \end{bmatrix}$$

Proof. Let $\boldsymbol{x} \in \mathbb{L}$ be a length L function vector, then using the results of Propositions 3.4 and 3.5 and Theorems 3.7 and 3.9, we find the coefficients of $\boldsymbol{\mathcal{Q}x}$ are given by

$$\mathbf{c}_{\boldsymbol{\mathcal{Q}}}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{c}_{\boldsymbol{y}^{\triangledown}} \\ \sum_{j=1}^{L-1} \mathbf{A}_j \mathbf{c}_{y_j^{\triangledown}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}\mathbf{c}_{\boldsymbol{x}^{\triangle}} \\ \sum_{j=1}^{L-1} \mathbf{A}_j \mathbf{c}_{x_j^{\triangle}} \end{bmatrix}.$$

□

The following corollary gives the coefficients of the length L functions that are generated in FSSA V-forecasting.

Corollary 3.11. *The* Ld*-dimensional vectors of coefficients corresponding to the* $\boldsymbol{w}_j$*'s given in equation* (37) *can be written as*

$$\mathbf{c}_{\boldsymbol{w}_j} = \begin{cases} \mathbf{c}_{\boldsymbol{x}_j^{\mathfrak{s}}}, & j = 1, \ldots, K \\ \mathbf{c}_{\boldsymbol{\mathcal{Q}}}(\boldsymbol{w}_{j-1}), & j = K+1, \ldots, K+M \end{cases}. \tag{39}$$

Now we offer some numerical studies that illustrate that our novel algorithms outperform the method of Hyndman and Ullah (2007) in FTS forecasting for periodic data.

3.4 Numerical Studies

In this section, we offer a simulation study and real data examples that showcase the advantage of our novel methods in forecasting periodic FTS. In the simulation study, we compare our approaches with the functional seasonal naive method and we also compare to the popular FTS forecasting technique of Hyndman and Ullah (2007), which we now call the competing method. We incorporate either seasonality, increasing trend, or both modes of variation in various simulation setups. We further illustrate the superior performance of our approaches in forecasting of a periodic FTS with a real data study where we compare our techniques with the competing method when the optimal parameters are selected for each algorithm applied to three different datasets. The three data sets considered are highly periodic call center data, moderately periodic remote sensing data, and

mortality rate data that has no periodic components. We note that when we state that we are using k eigentriples in forecasts, that means we use k left singular functions in the prediction.

3.4.1 Simulation Study

For the simulation study, we use a setup that is similar to that seen in Subsection 2.4.1 where varying cases will generate FTS that have only periodicity or only trend or both modes of variation. This particular setup utilizes FTS of lengths $N = 100, 200$ that are observed on $n = 100$ fixed, equidistant discrete points on the unit interval from the following model:

$$Y_t(s_i) = m_t(s_i) + X_t(s_i), \quad s_i \in [0,1], i = 1, \ldots, n, \text{ and } t = 1, \ldots, N.$$

We use a B-spline basis with 15 basis elements to smooth the discrete samplings of function curves. We have that $m_t(s)$ is the true underlying signal to be extracted and predicted where

$$m_t(s) = \kappa t + e^{s^2} \cos(2\pi\omega t) + \cos(4\pi s) \sin(2\pi\omega t),$$

and we allow the trend coefficient, κ, to take on values of 0 and 0.02 while the frequency, ω, takes on the values of 0 or 0.20 in varying setups. We also have that $X_t(s)$ is a stochastic term that follows a functional autoregressive model of order 1, FAR(1), defined by equation (29), equation (30), and equation (31). In this case, we choose γ_0 such that the norm of the operator in equation (31) takes on values of 0.25, 0.60, 0.90, and 0.95. We also consider the functions, $\varepsilon_t(s)$, to be independent trajectories of standard Brownian motion over $[0, 1]$. We compare our FSSA R-forecasting (FSSAR) method to our FSSA V-forecasting (FSSAV) algorithm with varying lags, L, of 10 and 20 and we compare both of these methods to the functional seasonal naive method (SNM) and the competing method (H & U). We choose the cases for the lag parameter in accordance with SSA literature where L is often chosen to be a multiple of the periodicity in the data, see Golyandina et al. (2001). In addition, we choose the optimal number of eigentriples to perform reconstructions and forecasts in FSSA-based models for each combination of κ and ω (two when $\kappa = 0$, $\omega = 0.20$, three when $\kappa = 0.02$, $\omega = 0.20$, and one when $\kappa = 0.02$, $\omega = 0$). We have that the maximum number of left singular functions chosen in the FSSA-based predictions is three (when $\kappa = 0.02$ and $\omega = 0.20$) and due to this, we choose three functional principal components to perform all forecasts in the method of Hyndman and Ullah (2007) for the simulations.

We expect that since our methods incorporate periodicity into the left singular functions that are used in the forecast, our techniques will perform better than the competing approach in predicting periodic true signals. To perform the comparison, we leverage a rolling forecast where the size of our training set, O, takes on values of 60 and 80. We denote the forecast of the t^{th} function evaluated at $s_i \in [0,1]$ with $\hat{Y}_t(s_i)$, which allows us to define the root mean square error of

$$RMSE = \sqrt{\frac{1}{(N-O)\times n}\sum_{t=1}^{N-O}\sum_{i=1}^{n}\left(\hat{Y}_{t+O}(s_i) - m_{t+O}(s_i)\right)^2}.$$

From here, we replicate the result of each simulation setup combination 100 times and report the mean of the $RMSE$'s in Figure 9.

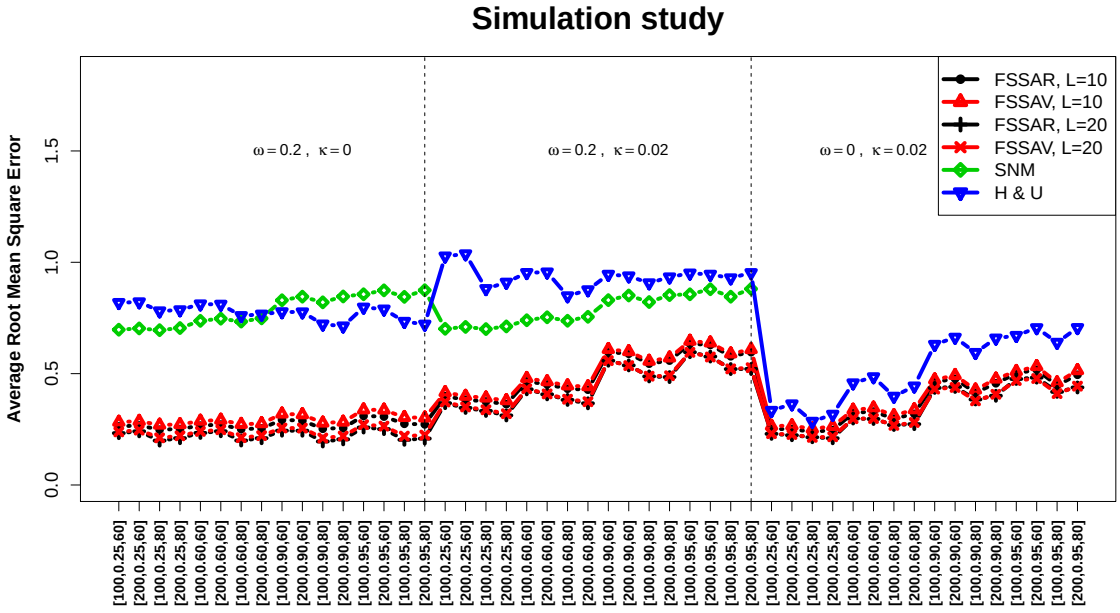

Figure 9: Simulation study where horizontal axis entries have form of $\left[N, \|\Psi\|_{\mathcal{S}}^2, O\right]$ and the vertical axis is average prediction root mean square error for each simulation setup replicated 100 times.

We find that our novel approaches consistently outperform the SNM approach and the competing methodology for all setups. These results suggest the superior performance of our novel methodologies in predicting the true underlying signal when periodicity is present in the data and how our methods are still competitive for data that have no oscillatory elements. For a visuanimation that displays a sample of the simulation setups and some of the graphical results, see Section B.1 of the Appendix.

We note that principal components that account for small variation in the data can still play a big role in improving predictions in principal component regression, see Jolliffe (1982) and references therein. As a result, we may have to use more than three functional principal components when applying the method of Hyndman and Ullah (2007) in FTS forecasting however, the fact that our methods outperform the competing approach with such few eigentriples used is a bonus. Now in the real data study, we showcase that our methodologies still outperform the functional principal component-based technique of Hyndman and Ullah (2007), when applied to highly periodic FTS, even when the optimal number of functional principal components are chosen for the competing method.

3.4.2 Real Data Study

We first introduce three real FTS each with varying levels of periodicity and trend being present. The first FTS we introduce is the call center data shown in Figure 10(A). The second is the NDVI density curves shown in Figure 10(B). Both of the aforementioned datasets are presented and discussed in Chapter 2. The final data set is comprised of 97 functional observations representative of morality rates of Swedish males between ages 0 and 100 between 1899 and 1995 shown in Figure 10(C) (University of California at Berkely (USA) and Max Planck Institue for Demographic Research, 2020). We note that for all subplots in Figure 10, warm colors, such as red and yellow, are indicative of curves observed on earlier dates while cool colors, such as blue and purple, are indicative of curves observed on later dates.

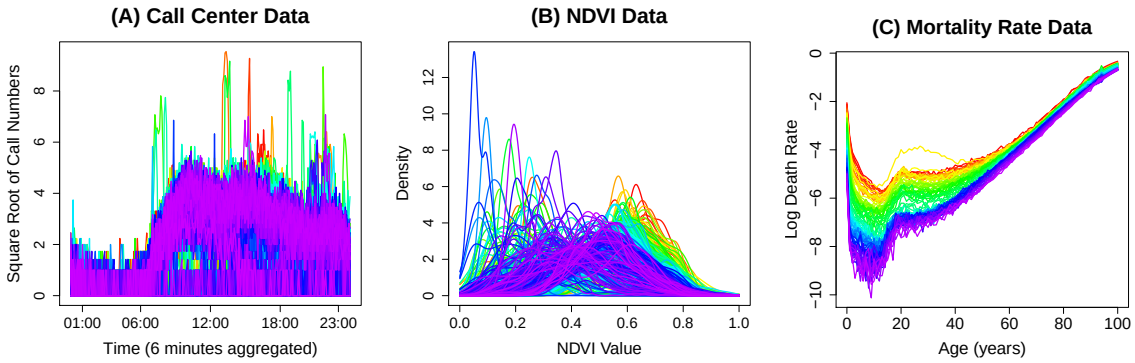

Figure 10: (A): The call center data; (B): The NDVI densities data; (C): The Swedish mortality rate data; The subplots were generated using the **rainbow** package (Shang and Hyndman, 2019).

Using Figure 10, we can infer existence of periodic and trend behaviors present in each data set. Figure 10(A) clearly shows periodic behavior with no trend as all warmer colors are covered with

cooler colors. This strong, weekly periodicity for the call center data is also uncovered in Section A.2 of the Appendix. We recall that in Subsection 2.4.2 that there exists periodic and trend behaviors in the NDVI densities shown in Figure 10(B). Finally, we see a strong decreasing trend with no periodicity in the mortality rate data in Figure 10(C). As demonstrated in the simulation study, our novel algorithms appear to perform better in forecasting as compared to the competing method when a FTS is periodic. As such, we expect our novel approaches to perform better than the competing method for the highly periodic call center data and we expect a closer competition for the NDVI densities and mortality rate data.

As according to Section A.2 of the Appendix, for the FSSA-based prediction algorithms applied to call center data, we select $L = 28$ and the first seven eigentriples to build our deterministic FTS in reconstruction and to perform the forecast. Also according to Subsection 2.4.2, for the FSSA-based algorithms applied to NDVI data, we select $L = 45$ and the first four eigentriples to build the FTS and perform the forecast. Since there are no periodic elements present in the mortality rate data, we cannot use the standard rule of leveraging a lag that is a multiple of the periodicity in the FTS for the FSSA-based algorithms. In addition, we must select the best number of eigentriples to reconstruct the deterministic signal and to do the prediction given some lag L for the mortality rate data. We also must select the optimal number of functional principal components to use in the forecasts while applying the method of Hyndman and Ullah (2007) to each of the three data sets. To achieve these goals, we look towards cross-validation via a rolling forecast. To achieve these goals, we define prediction root mean square error by

$$prRMSE = \sqrt{\frac{1}{(N-O) \times n} \sum_{t=1}^{N-O} \sum_{i=1}^{n} \left(\hat{Y}_{t+O}(s_i) - Y_{t+O}(s_i) \right)^2} \tag{40}$$

where $\hat{Y}_{t+O}$ is the prediction of observation Y_{t+O}, s_i is a point in the domain of each FTS, n are the number of sampling points, N is the length of the FTS, and O is the training set size. We perform rolling forecasts with training set sizes of 308, 403, and 50 for the call center, NDVI, and mortality data respectively (giving us testing set sizes of 57, 45, and 47 respectively), and we estimate $prRMSE$ as according to equation (40) for varying choices of parameters. We find that a lag of 24 and using the first eigentriple to perform reconstruction and forecasting in the FSSA-based algorithms applied to the mortality rate data minimize equation (40) for that data set. In terms of choosing the optimal number of functional principal components to perform the forecast in the method of Hyndman and Ullah (2007), we find choosing the first 33 functional principal components, the first eight functional

principal components, and the first three functional principal components for the call center, NDVI, and mortality rate data respectively minimize equation (40). We report the prediction root mean square errors for each optimal model in Table 3.

Methodology	Call Center	NDVI	Mortality
FSSA R-Forecasting	0.569	0.827	0.236
FSSA V-Forecasting	0.567	0.824	0.227
Method of Hyndman and Ullah (2007)	0.855	0.818	0.143

Table 3: $prRMSE$'s of FSSA R-forecasting, FSSA V-forecasting, and the competing method of Hyndman and Ullah (2007) for various datasets.

As seen in Table 3, our novel methodologies outperform for the highly periodic call center data. The competing method outperforms our novel techniques for the NDVI data which has a mixture of trend and periodic components. Finally, we see that the competing method performs better for the mortality rate data which contains no periodic components and only a decreasing trend.

3.5 Discussion

In this chapter, we developed efficient nonparametric FTS forecasting techniques that include periodicity into the basis elements. We compared our approaches with the competing method of Hyndman and Ullah (2007) and found our techniques are superior in forecasting highly periodic FTS since the FSSA routine captures time-dependency in the basis elements. As a result researchers can use our methodologies in order to obtain accurate and informative predictions of periodic stochastic processes.

CHAPTER 4: MULTIVARIATE FUNCTIONAL SINGULAR SPECTRUM ANALYSIS OVER DIFFERENT DIMENSIONAL DOMAINS

In this chapter, we develop MFSSA over different dimensional domains which is the functional extension of MSSA. The MFSSA algorithm is a decomposition routine for multivariate functional data that are time-dependent and can be comprised of functions over any finite dimensional domain. In the following, we provide all necessary theoretical details supporting the work as well as the computer implementation strategy needed for the algorithm. We provide a simulation study showcasing the superior performance in reconstruction accuracy of a MFTS signal found using MFSSA as compared to other approaches and we give a real data study showing how MFSSA enriches analysis using intraday temperature curves and smoothed remote sensing images of vegetation. MFSSA is available for use through the **Rfssa** R package.

4.1 Introduction

In practice it is often the case that many variables are observed as a result of a single stochastic process and investigation of time series components can be made richer by performing a multivariate analysis of these vector observations. The MSSA algorithm, reviewed in Chapter 1, is a technique that has seen success over its univariate SSA counterpart in decomposing a multidimensional time series into components if the covariates are moderately correlated (Golyandina et al., 2015). As mentioned in Subsection 1.3.3, the MSSA algorithm also has been developed in two different approaches of vertical MSSA (VMSSA) and horizontal MSSA (HMSSA) where VMSSA involves the vertical stacking of univariate Hankel trajectory matrices while HMSSA works with the horizontal stacking of the same elements (Hassani and Mahmoudvand, 2018). These two approaches to MSSA have both been extended to allow for prediction of multidimensional time series via recurrent and vector forecasting algorithms where the extensions for the horizontal approach are presented in Golyandina et al. (2015) and extensions for both approaches are presented in Hassani and Mahmoudvand (2018). Over the course of the last 15 years, MSSA has seen significant success in various areas of application see Golyandina et al. (2015); Groth and Ghil (2011); Hassani et al. (2019); Silva et al. (2018).

Multivariate functional data are observed when a stochastic process gives rise to multiple different functions over possibly different dimensional domains. Multivariate functional principal component analysis was developed so that more than one variable of functional subjects could be included in the analysis. Chiou et al. (2014) extended multivariate functional principal component analysis method to include a normalized approach which accounts for differences in degrees of variability in the

covariates as well as differences in units. The multivariate functional principal component analysis algorithm was further extended by Happ and Greven (2018) to account for different dimensional domains so that one could perform dimension reduction on multivariate functional data that might be comprised of curves, surfaces, or any other finite dimensional domain altogether. A primary assumption of multivariate functional principal component analysis is that the functional data are time-independent. With the goal of performing decomposition on a MFTS, one might conjecture to use FSSA on the covariates independently of one another but this fails to capture any cross-correlations between variables. MFSSA provides us a way to perform decomposition of a MFTS while capturing these cross-correlations to further enrich analysis and strengthen reconstruction accuracy of the true signal. In addition, MFSSA is developed to handle functions taken over any finite dimensional domain. This can allow the user to explore relationships between time-dependent curves, smoothed images, or any other function whose domain is finite dimensional.

The rest of the chapter is organized as follows. We present the functional extension of MSSA known as MFSSA in Section 4.2. We then offer the techniques used to implement MFSSA in Section 4.3. Next we show how one can generalize MSSA into MFSSA by developing both horizontal MFSSA (HMFSSA) and vertical MFSSA (VMFSSA) in Section 4.4. We also show in Section 4.4 that VMFSSA solves the same problem as MFSSA using an isomorphic mapping. In Section 4.5, we offer a simulation study that shows that MFSSA outperforms all other known methods in terms of reconstruction accuracy of a MFTS signal. Also in Section 4.5, we give a real data study where we use weather station intraday temperature curves and smoothed remote sensing images in a bivariate analysis to explore some of the more interesting qualities of MFTS data through the use of MFSSA. We end with a discussion of results in Section 4.6.

4.2 Theoretical Foundations of Multivariate Functional Singular Spectrum Analysis

The mathematical foundations in the following are used throughout the chapter and form the theoretical backbone of the MFSSA algorithm.

Preliminaries and Notations

For each $j = 1, \ldots, p$, consider an m_j-dimensional domain, T_j, to be a compact subset of $\mathbb{R}^{m_j}$, and let $\mathbb{F}_j = L^2(T_j)$ be the Hilbert space of square integrable function vectors defined on T_j. From hereafter, we define the Hilbert space, $\mathbb{H}$, to be the Cartesian product vector space given by $\mathbb{H} = \mathbb{F}_1 \times \cdots \times \mathbb{F}_p$, where each multivariate function vector, $\vec{x} \in \mathbb{H}$, can be denoted by the p-tuple $\left(x^{(1)}, \ldots, x^{(p)}\right)$. Note

that $\mathbb{H}$ is equipped with inner product given by

$$\langle \vec{x}, \vec{y} \rangle_{\mathbb{H}} = \sum_{j=1}^{p} \left\langle x^{(j)}, y^{(j)} \right\rangle_{\mathbb{F}_j} = \sum_{j=1}^{p} \int_{T_j} x^{(j)}(s_j)\, y^{(j)}(s_j)\, ds_j, \quad s_j \in T_j,$$

for some multivariate function vectors, $\vec{x}, \vec{y} \in \mathbb{H}$. From hereafter, we specify a MFTS of length N as $\mathbf{y}_N = (\vec{y}_1, \ldots, \vec{y}_N)$, where $\vec{y}_i \in \mathbb{H}$.

Similarly, for a given $L \in \mathbb{N}$, $\mathbb{H}^L$ is a function vector space that is built from the Cartesian product of L copies of $\mathbb{H}$, and each length L multivariate function vector, $\boldsymbol{x} \in \mathbb{H}^L$, can be denoted by the L-tuple $(\vec{x}_1, ..., \vec{x}_L)$. Clearly $\mathbb{H}^L$ is a Hilbert space with respect to the inner product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{H}^L} = \sum_{i=1}^{L} \langle \vec{x}_i, \vec{y}_i \rangle_{\mathbb{H}}, \quad \text{for } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}^L.$$

Next we define $\mathbb{H}^{L\times K}$ to be the space spanned by linear operators, $\boldsymbol{\mathcal{V}} : \mathbb{R}^K \to \mathbb{H}^L$, specified by multivariate function vectors, $(\vec{v}_{i,k})_{i=1,\ldots,L}^{k=1,\ldots,K} \in \mathbb{H}$, as

$$\boldsymbol{\mathcal{V}}(\mathbf{a}) = \left(\sum_{k=1}^{K} a_k \vec{v}_{1,k}, \ldots, \sum_{k=1}^{K} a_k \vec{v}_{L,k} \right), \qquad \mathbf{a} = [a_1, a_2, \ldots, a_K]^\top \in \mathbb{R}^K.$$

For two operators $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{Z}} \in \mathbb{H}^{L\times K}$, the Frobenius inner product can be defined as

$$\langle \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{Z}} \rangle_F = \sum_{i=1}^{L} \sum_{k=1}^{K} \langle \vec{v}_{i,k}, \vec{z}_{i,k} \rangle_{\mathbb{H}},$$

which induces the Frobenius norm given by $\|\boldsymbol{\mathcal{V}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{V}} \rangle_F}$. We denote $\mathbb{H}_H^{L\times K}$ to be the Hankel subspace of $\mathbb{H}^{L\times K}$ such that for any operator, $\widetilde{\boldsymbol{\mathcal{V}}} = \left(\vec{\tilde{v}}_{i,k}\right) \in \mathbb{H}_H^{L\times K}$, there exists a multivariate function vector, $\vec{g}_u \in \mathbb{H}$, such that $\|\vec{\tilde{v}}_{i,k} - \vec{g}_u\|_{\mathbb{H}} = 0$ where $u = i + k$.

4.2.1 Multivariate Functional Singular Spectrum Analysis Algorithm

Similar to other SSA-based algorithms, MFSSA consists of four steps: Embedding, Decomposition, Grouping, and Reconstruction. In the first step of embedding, we establish a so-called multivariate trajectory operator whose range is built from elements that describe MFTS behavior over sub-intervals of time.

Step 1. Embedding

As one may note the columns of a univariate trajectory matrix, as given in equation (8), are the corresponding L-lagged vectors for the j^{th} variable. Therefore a trajectory matrix can be seen as a linear operator from $\mathbb{R}^K$ to the space of linear combinations of the L-lagged vectors. We use this as a motivation to introduce the trajectory operator for FSSA in Chapter 2.

In a similar fashion, we define the L-lagged multivariate function vectors in $\mathbb{H}^L$ of the form

$$\boldsymbol{x}_k = (\vec{y}_k, \vec{y}_{k+1}, \ldots, \vec{y}_{k+L-1}), \quad k = 1, \ldots, K \tag{41}$$

which correspond with the MFTS, $\mathbf{y}_N$. We define a linear operator, specified with $\boldsymbol{x}_k$'s, to obtain the multivariate trajectory operator, $\boldsymbol{\mathcal{X}} : \mathbb{R}^K \to \mathbb{H}^L$. As such, for some vector $\mathbf{a} = [a_1, a_2, \ldots, a_K]^\top \in \mathbb{R}^K$, we have

$$\boldsymbol{\mathcal{X}}(\mathbf{a}) = \sum_{k=1}^{K} a_k \boldsymbol{x}_k. \tag{42}$$

We define $R(\boldsymbol{\mathcal{X}}) = \text{span}\{\boldsymbol{x}_k\}_{k=1}^K$ to be the range of the operator $\boldsymbol{\mathcal{X}}$ and clearly, the rank of the multivariate trajectory operator is, $r \in \mathbb{N}$, where $1 \leq r \leq K$. This step of embedding can also be viewed as applying the invertible transformation, $\boldsymbol{\mathcal{T}} : \mathbb{H}^N \to \mathbb{H}_H^{L \times K}$, such that

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}}(\mathbf{y}_N). \tag{43}$$

In the following Proposition, we establish some useful properties of $\boldsymbol{\mathcal{X}}$ and we define it's adjoint.

Proposition 4.1. *The operator given in equation* (42) *is a bounded and linear operator with adjoint* $\boldsymbol{\mathcal{X}}^* : \mathbb{H}^L \to \mathbb{R}^K$ *where for some length L multivariate function vector,* $\boldsymbol{z} \in \mathbb{H}^L$*, we obtain the following* K*-dimensional vector*

$$\boldsymbol{\mathcal{X}}^* \boldsymbol{z} = \left[\langle \boldsymbol{x}_1, \boldsymbol{z} \rangle_{\mathbb{H}^L}, \langle \boldsymbol{x}_2, \boldsymbol{z} \rangle_{\mathbb{H}^L}, \ldots, \langle \boldsymbol{x}_K, \boldsymbol{z} \rangle_{\mathbb{H}^L}\right]^\top \in \mathbb{R}^K.$$

Proof. Notice that since $\boldsymbol{\mathcal{X}}$ is a finite rank operator and thus, $\boldsymbol{\mathcal{X}}$ is bounded and linear as a result. Now let $\boldsymbol{z} \in \mathbb{H}^L$, then

$$\langle \boldsymbol{\mathcal{X}}\mathbf{a}, \boldsymbol{z} \rangle_{\mathbb{H}^L} = \sum_{k=1}^{K} a_k \langle \boldsymbol{x}_k, \boldsymbol{z} \rangle_{\mathbb{H}^L} = \mathbf{a}^\top \boldsymbol{\mathcal{X}}^* \boldsymbol{z} = \langle \mathbf{a}, \boldsymbol{\mathcal{X}}^* \boldsymbol{z} \rangle_{\mathbb{R}^K}, \quad \mathbf{a} \in \mathbb{R}^K$$

and we have that $\boldsymbol{\mathcal{X}}^*$ is the adjoint of $\boldsymbol{\mathcal{X}}$. □

Now we discuss the second step of MFSSA where we extract out time-dependent modes of variation from $\boldsymbol{\mathcal{X}}$.

Step 2. Decomposition

Since $\boldsymbol{\mathcal{X}}$ is a finite rank operator, one may employ Theorem 7.6 of Weidmann (1980) and obtain a multivariate FSVD (MFSVD) for $\boldsymbol{\mathcal{X}}$ as

$$\boldsymbol{\mathcal{X}}(\mathbf{a}) = \sum_{i=1}^{r} \sqrt{\lambda_i} \langle \mathbf{v}_i, \mathbf{a} \rangle_{\mathbb{R}^K} \boldsymbol{\psi}_i = \sum_{i=1}^{r} \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i (\mathbf{a}) \tag{44}$$

where $\otimes$ stands for the tensor (outer) product. From hereafter, $\{\sqrt{\lambda_i}\}_{i=1}^r$ are the singular values, $\{\mathbf{v}_i\}_{i=1}^r$ are the orthonormal right singular vectors spanning an r-dimensional subspace of $\mathbb{R}^K$, and $\{\boldsymbol{\psi}_i\}_{i=1}^r$ are the orthonormal left singular functions spanning an r-dimensional subspace of $\mathbb{H}^L$. We call the set, $\{\sqrt{\lambda_i}, \boldsymbol{\psi}_i, \mathbf{v}_i\}$, as the i^{th} eigentriple of $\boldsymbol{\mathcal{X}}$. Now we define the rank one elementary operators built from the eigentriples, $\boldsymbol{\mathcal{X}}_i : \mathbb{R}^K \to \mathbb{H}^L$, given by $\boldsymbol{\mathcal{X}}_i = \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i$ for $i = 1, \ldots, r$. It is easy to see that $\boldsymbol{\mathcal{X}} = \sum_{i=1}^r \boldsymbol{\mathcal{X}}_i$. The following Proposition establishes equalities between $\boldsymbol{\mathcal{X}}$ and eigentriple elements.

Proposition 4.2. *Let $\{\sqrt{\lambda_i}, \boldsymbol{\psi}_i, \mathbf{v}_i\}$ be the i^{th} eigentriple of $\boldsymbol{\mathcal{X}}$, $i = 1, \ldots, r$. The following holds:*

$$\boldsymbol{\psi}_i = \left(\sqrt{\lambda_i}\right)^{-1} \boldsymbol{\mathcal{X}} \mathbf{v}_i, \quad \mathbf{v}_i = \left(\sqrt{\lambda_i}\right)^{-1} \boldsymbol{\mathcal{X}}^* \boldsymbol{\psi}_i.$$

Proof. Recall that a MFSVD of $\boldsymbol{\mathcal{X}}$ gives us a set of orthonormal vectors, $\{\mathbf{v}_i\}_{i=1}^r$, and a set of orthonormal left singular functions, $\{\boldsymbol{\psi}_i\}_{i=1}^r$. We find that

$$\boldsymbol{\mathcal{X}}(\mathbf{v}_i) = \sum_{j=1}^{r} \sqrt{\lambda_j} \left(\mathbf{v}_i^\top \mathbf{v}_j\right) \boldsymbol{\psi}_j = \sqrt{\lambda_i} \boldsymbol{\psi}_i.$$

From here, we have that $\boldsymbol{\psi}_i = \left(\sqrt{\lambda_i}\right)^{-1} \boldsymbol{\mathcal{X}}(\mathbf{v}_i)$. By Theorem 7.6 of Weidmann (1980), we have that the operator, $\boldsymbol{\mathcal{X}}^*$, has a MFSVD with the same eigentriples as $\boldsymbol{\mathcal{X}}$ and we obtain the following

$$\boldsymbol{\mathcal{X}}^*(\boldsymbol{\psi}_i) = \sum_{j=1}^{r} \sqrt{\lambda_j} \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle_{\mathbb{H}^L} \mathbf{v}_j = \sqrt{\lambda_i} \mathbf{v}_i$$

which implies that $\mathbf{v}_i = \left(\sqrt{\lambda_i}\right)^{-1} \boldsymbol{\mathcal{X}}^*(\boldsymbol{\psi}_i)$.

□

Now we present step three of the MFSSA algorithm where we group together eigentriples that describe similar MFTS behavior.

Step 3. Grouping

The grouping step of MFSSA follows the same manner as the grouping step of MSSA. We partition the set of indices, $\{1, 2, \ldots, r\}$, into m disjoint subsets, $\{I_1, I_2, \ldots, I_m\}$, such that for $q = 1, \ldots m$, the operator, $\boldsymbol{\mathcal{X}}_{I_q} : \mathbb{R}^K \to \mathbb{H}^L$, is defined by $\boldsymbol{\mathcal{X}}_{I_q} = \sum_{i \in I_q} \boldsymbol{\mathcal{X}}_i$. As such, we have that

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{I_1} + \boldsymbol{\mathcal{X}}_{I_2} + \cdots + \boldsymbol{\mathcal{X}}_{I_m}.$$

Similar to other forms of SSA, exploratory plots, such as scree plots, paired-plots, w-correlation plots, and others can be developed to determine how to obtain the m disjoint groups. In addition, we also have that each operator, $\boldsymbol{\mathcal{X}}_{I_q}$, corresponds with some component of the original MFTS such as mean, periodic, trend, or other behaviors. In the next step, our goal is to extract a MFTS from each $\boldsymbol{\mathcal{X}}_{I_q}$ that describes on of these modes of variation.

Step 4. Reconstruction

Let $\boldsymbol{\mathcal{Y}} \in \mathbb{H}^{L \times K}$ be an operator defined by $\boldsymbol{\mathcal{Y}} = (\vec{y}_{i,j})$. Since $\mathbb{H}_H^{L \times K}$ is a closed subspace of $\mathbb{H}^{L \times K}$, we have by Theorem 3.2.3 of Shalit (2017) that there exists a unique Hankel operator, $\widetilde{\boldsymbol{\mathcal{Y}}} = \left(\vec{\tilde{y}}_{i,j}\right) \in \mathbb{H}_H^{L \times K}$, such that

$$\|\boldsymbol{\mathcal{Y}} - \widetilde{\boldsymbol{\mathcal{Y}}}\|_F^2 \leq \|\boldsymbol{\mathcal{Y}} - \widetilde{\boldsymbol{\mathcal{Z}}}\|_F^2,$$

for any Hankel operator, $\widetilde{\boldsymbol{\mathcal{Z}}} \in \mathbb{H}_H^{L \times K}$. Define the orthogonal projection, $\boldsymbol{\Pi}_H : \mathbb{H}^{L \times K} \to \mathbb{H}_H^{L \times K}$, such that we have $\widetilde{\boldsymbol{\mathcal{Y}}} = \boldsymbol{\Pi}_H \boldsymbol{\mathcal{Y}}$. We achieve the projection, $\boldsymbol{\Pi}_H$, by using Lemma 2.1 and the resulting diagonal averaging technique given by

$$\vec{\tilde{y}}_{i,k} = \frac{1}{n_u} \sum_{(n,m):n+m=u} \vec{y}_{n,m}, \tag{45}$$

where n_u is the number of (n, m) pairs such that $n + m = u$. With $\boldsymbol{\Pi}_H$, we have that $\widetilde{\boldsymbol{\mathcal{X}}}_{I_q} = \boldsymbol{\Pi}_H \boldsymbol{\mathcal{X}}_{I_q}$ for $q = 1, \ldots, m$. We then employ the inverse of $\boldsymbol{\mathcal{T}} : \mathbb{H}^N \to \mathbb{H}_H^{L \times K}$, from equation (43), to obtain the following formula for the reconstruction

$$\tilde{\mathbf{y}}_N^q = \boldsymbol{\mathcal{T}}^{-1} \widetilde{\boldsymbol{\mathcal{X}}}_{I_q}.$$

Similar to other types of SSA, we have that each $\tilde{\mathbf{y}}_N^q$ is a MFTS that describes a component of the original signal, $\mathbf{y}_N$, such as mean behavior, periodicity, trend, etc. Even though MFSSA is a model-free procedure, it is ideal that $\mathbf{y}_N$ have additive components which implies that the MFTS is separable. In the following, we introduce the idea of separability for MFTS.

4.2.2 Separability

Let $\mathbf{x}_N = \mathbf{y}_N + \mathbf{z}_N$ where $\mathbf{y}_N$ and $\mathbf{z}_N$ are MFTS of length N. We define the weighted-covariance between the MFTS as

$$\langle \mathbf{y}_N, \mathbf{z}_N \rangle_w = \sum_{j=1}^{p} \sum_{i=1}^{N} w_i \left\langle y_i^{(j)}, z_i^{(j)} \right\rangle_{\mathbb{F}_j},$$

where $w_i = \min\{i, L, N-i+1\}$. The definition of the norm, $\|\mathbf{y}_N\|_w$, the discussion on w-orthogonality of $\mathbf{y}_N$ and $\mathbf{z}_N$, and the definition of the w-correlation matrix is straightforward from the FSSA discussion on separability and we may use these tools to help in the grouping stage of MFSSA. Now we discuss the computer implementation strategy.

4.3 Computer Implementation Strategy

In the following, we describe the basis that is used to represent a multivariate function vector in $\mathbb{H}$. We then use the basis for the multivariate function vectors to construct the basis used to build the L-lagged multivariate function vectors in $\mathbb{H}^L$, and the implementation of the multivariate trajectory operator, $\boldsymbol{\mathcal{X}}$.

Let $\{\nu_i^{(j)}\}_{i\in\mathbb{N}}$ be the collection of linearly independent basis function vectors in the function vector space, $\mathbb{F}_j$ for $j = 1, ..., p$. Each observation in $\mathbb{F}_j$ can be projected onto the subspace $F_j = \text{span}\{\nu_i^{(j)}\}_{i=1}^{d_j}$ where d_j is a positive integer that can be determined by a variety of techniques like cross-validation. To this end, each function vector, $y_i^{(j)} \in F_j$, can be uniquely expressed as

$$y_i^{(j)} = \sum_{k=1}^{d_j} c_{i,k}^{(j)} \nu_k^{(j)}, \quad i = 1, ..., N, \quad c_{i,k}^{(j)} \in \mathbb{R}.$$

Now we set $d_0 = 0$, $d = \sum_{j=0}^{p} d_j$, and define the Hilbert space, $\mathbb{H}_d = F_1 \times \ldots \times F_p \subset \mathbb{H}$.

For each $q \in \{1, \ldots, d\}$, there exist a unique $j_q \in \{1, \ldots, p\}$ such that $\sum_{i=0}^{j_q-1} d_i + 1 \leq q \leq \sum_{i=0}^{j_q} d_i$. Now consider the multivariate function vector, $\vec{\nu}_q \in \mathbb{H}_d$, which is the zero function in all components except the j_q^{th} element, which is $\nu_{\ell_q}^{(j_q)}$, where $\ell_q = q - \sum_{i=0}^{j_q-1} d_i$. We expand on some interesting properties of the multivariate function vectors, $\vec{\nu}_q$'s, in the following Lemma.

Lemma 4.3. *The following holds:*

i) Each multivariate function vector, $\vec{y}_i \in \mathbb{H}_d$, can be uniquely represented as a linear combination of $\vec{\nu}_q$'s such that

$$\vec{y}_i = \sum_{q=1}^{d} c_{i,\ell_q}^{(j_q)} \vec{\nu}_q, \quad i = 1, ..., N.$$

ii) The set $\{\vec{\nu}_q\}_{q=1}^{d}$ is a basis system of $\mathbb{H}_d$.

Proof. We prove the two parts of this lemma in the following.

i) Let $M_{j_q} = \sum_{i=0}^{j_q} d_i$, then we obtain the following multivariate function vectors of $\mathbb{H}_d$.

$$\begin{aligned}
\vec{y}_i^{(1)} &= \left(y_i^{(1)}, 0, \ldots, 0\right) = \sum_{q=1}^{d_1} c_{i,\ell_q}^{(1)} \vec{\nu}_q \\
\vec{y}_i^{(2)} &= \left(0, y_i^{(2)}, 0, \ldots, 0\right) = \sum_{q=d_1+1}^{d_1+d_2} c_{i,\ell_q}^{(2)} \vec{\nu}_q \\
&\vdots \\
\vec{y}_i^{(j_q)} &= \left(0, \ldots, 0, y_i^{(j_q)}, 0, \ldots, 0\right) = \sum_{q=M_{j_q-1}+1}^{M_{j_q}} c_{i,\ell_q}^{(j_q)} \vec{\nu}_q \\
&\vdots \\
\vec{y}_i^{(p)} &= \left(0, \ldots, 0, y_i^{(p)}\right) = \sum_{q=M_{j_p-1}+1}^{d} c_{i,\ell_q}^{(p)} \vec{\nu}_q.
\end{aligned}$$

From this, we find that any multivariate function vector, $\vec{y}_i \in \mathbb{H}_d$, can be expressed as

$$\begin{aligned}
\vec{y}_i &= \left(y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(j_q)}, \ldots, y_i^{(p)}\right) \\
&= \vec{y}_i^{(1)} + \vec{y}_i^{(2)} + \cdots + \vec{y}_i^{(j_q)} + \cdots + \vec{y}_i^{(p)} = \sum_{q=1}^{d} c_{i,\ell_q}^{(j_q)} \vec{\nu}_q.
\end{aligned}$$

ii) This part of the proof is a direct consequence of the proof of part *i)*

□

Now for each $k \in \{1, \ldots, Ld\}$, one can see that there exists unique $q_k \in \{1, \ldots, d\}$ and $r_k \in \{1, \ldots, L\}$ such that $k = (q_k - 1)L + r_k$. Consider $\boldsymbol{\phi}_k$ as a length L multivariate function vector that is zero in all components, except the r_k-th component, which is $\vec{\nu}_{q_k}$. The following Lemma now defines a

linearly independent basis for the L-lagged multivariate function vectors, $\boldsymbol{x}_j$'s, where the basis is defined using the $\vec{\nu}_q$'s.

Lemma 4.4. *The sequence, $\{\boldsymbol{\phi}_k\}_{k=1}^{Ld}$, is a basis system for $\mathbb{H}_d^L$, where $\mathbb{H}_d^L$ is the Cartesian product of L copies of $\mathbb{H}_d$.*

Proof. The proof of this Lemma is almost identical to the proof of Lemma 2.6 and holds without loss of generality. □

Using Lemma 4.4, one may define a linear operator $\boldsymbol{\mathcal{P}} : \mathbb{R}^{Ld} \to \mathbb{H}_d^L$, specified with $\boldsymbol{\phi}_k$'s, such that any length L multivariate function vector, $\boldsymbol{x} \in \mathbb{H}_d^L$, can be written as

$$\boldsymbol{x} = \sum_{i=1}^{Ld} b_i \boldsymbol{\phi}_i = \boldsymbol{\mathcal{P}}(\mathbf{b}),$$

where $\mathbf{b} \in \mathbb{R}^{Ld}$. Similar to equation (41), one may define the L-lagged multivariate function vectors for the MFTS, $\mathbf{y}_N$, as $\boldsymbol{x}_k = (\vec{y}_k, \vec{y}_{k+1}, \ldots, \vec{y}_{k+L-1}) \in \mathbb{H}_d^L$, where $k = 1, \ldots, K$. Therefore the associated multivariate trajectory operator, given in equation (42), is defined as $\boldsymbol{\mathcal{X}} : \mathbb{R}^K \to \mathbb{H}_d^L$. The next Lemma gives the tools necessary to implement the L-lagged multivariate function vectors and $\boldsymbol{\mathcal{X}}$ using the $\boldsymbol{\phi}_i$'s.

Lemma 4.5. *The following holds:*

i) The corresponding coefficient vector of the k^{th} L-lagged multivariate function vector, $\boldsymbol{x}_k$, with respect to the operator $\boldsymbol{\mathcal{P}}$ is

$$\mathbf{b}_k = \left[c_{k,\ell_1}^{(j_1)}, \ldots, c_{k+L-1,\ell_1}^{(j_1)}, c_{k,\ell_2}^{(j_2)}, \ldots, c_{k+L-1,\ell_2}^{(j_2)}, \ldots, c_{k+L-1,\ell_d}^{(j_d)}\right]^\top \in \mathbb{R}^{Ld}.$$

ii) For any vector $\mathbf{a} \in \mathbb{R}^K$, we have that $\boldsymbol{\mathcal{X}}(\mathbf{a}) = \boldsymbol{\mathcal{P}}(\mathbf{B}\mathbf{a})$, where $\mathbf{B} = [b_{i,k}]_{i=1,\ldots,Ld}^{k=1,\ldots,K} = [\mathbf{b}_1 \mathbf{b}_2 \cdots \mathbf{b}_K]_{Ld \times K}$ is a matrix, and $b_{i,k}$ is the i^{th} component of the Ld-dimensional vector, $\mathbf{b}_k$.

Proof. We prove the two parts of this Lemma in the following

i) Let $M_{j_q} = \sum_{i=0}^{j_q} d_i$, then we obtain the following length L multivariate function vectors of $\mathbb{H}_d^L$.

$$\boldsymbol{x}_k^{(1)} = \begin{pmatrix} \vec{y}_k^{(1)} \\ \vec{y}_{k+1}^{(1)} \\ \vdots \\ \vec{y}_{k+L-1}^{(1)} \end{pmatrix} = \begin{pmatrix} \vec{y}_k^{(1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vec{y}_{k+1}^{(1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vec{y}_{k+L-1}^{(1)} \end{pmatrix} = \sum_{i=1}^{Ld_1} b_{i,k}\boldsymbol{\phi}_i$$

$$= \sum_{q_i=1}^{d_1} \sum_{r_i=1}^{L} c_{k+r_i-1,\ell_{q_i}}^{(1)} \boldsymbol{\phi}_i$$

$$\boldsymbol{x}_k^{(2)} = \begin{pmatrix} \vec{y}_k^{(2)} \\ \vec{y}_{k+1}^{(2)} \\ \vdots \\ \vec{y}_{k+L-1}^{(2)} \end{pmatrix} = \begin{pmatrix} \vec{y}_k^{(2)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vec{y}_{k+1}^{(2)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vec{y}_{k+L-1}^{(2)} \end{pmatrix} = \sum_{i=Ld_1+1}^{L(d_1+d_2)} b_{i,k}\boldsymbol{\phi}_i$$

$$= \sum_{q_i=d_1+1}^{d_1+d_2} \sum_{r_i=1}^{L} c_{k+r_i-1,\ell_{q_i}}^{(2)} \boldsymbol{\phi}_i$$

$$\vdots$$

$$\boldsymbol{x}_k^{(p)} = \begin{pmatrix} \vec{y}_k^{(p)} \\ \vec{y}_{k+1}^{(p)} \\ \vdots \\ \vec{y}_{k+L-1}^{(p)} \end{pmatrix} = \begin{pmatrix} \vec{y}_k^{(p)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vec{y}_{k+1}^{(p)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vec{y}_{k+L-1}^{(p)} \end{pmatrix} = \sum_{i=LM_{p-1}+1}^{Ld} b_{i,k}\boldsymbol{\phi}_i$$

$$= \sum_{q_i=M_{p-1}+1}^{d} \sum_{r_i=1}^{L} c_{k+r_i-1,\ell_{q_i}}^{(p)} \boldsymbol{\phi}_i.$$

As a result, we find that the k^{th} L-lagged multivariate function vector, $\boldsymbol{x}_k$, is given by $\boldsymbol{x}_k = \boldsymbol{x}_k^{(1)} + \boldsymbol{x}_k^{(2)} + \cdots + \boldsymbol{x}_k^{(p)} = \sum_{i=1}^{Ld} b_{i,k}\boldsymbol{\phi}_i$ and the components of $\mathbf{b}_k$ are found in Lemma 4.5.

ii)

$$\boldsymbol{\mathcal{X}}(\mathbf{a}) = \sum_{k=1}^{K} a_k \boldsymbol{x}_k = \sum_{i=1}^{Ld} \left(\sum_{k=1}^{K} b_{i,k} a_k \right) \boldsymbol{\phi}_i = \boldsymbol{\mathcal{P}}(\mathbf{Ba}), \quad \mathbf{a} \in \mathbb{R}^K.$$

□

Now that we have a way to represent our L-lagged multivariate function vectors and multivariate

trajectory operator using the $\boldsymbol{\phi}_i$'s, we leverage the following theorem to obtain the eigentriples of $\boldsymbol{\mathcal{X}}$.

Theorem 4.6. *Define the $Ld \times K$ matrix, $\mathbf{X} = \mathbf{G}^{1/2}\mathbf{B}$, where $\mathbf{G} = \left[\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle_{\mathbb{H}^L}\right]_{i,j=1}^{Ld}$ is the $Ld \times Ld$ Gram matrix. Denote the collection $\{\sqrt{\lambda_i}, \mathbf{v}_i, \mathbf{u}_i\}$ as the i^{th} eigentriple of $\mathbf{X}$. Now define $\boldsymbol{\psi}_i = \boldsymbol{\mathcal{P}}(\mathbf{G}^{-1/2}\mathbf{u}_i)$. The following holds:*

i) $\boldsymbol{\mathcal{X}}^*\boldsymbol{\psi}_i = \sqrt{\lambda_i}\mathbf{v}_i$

ii) $\boldsymbol{\mathcal{X}}\mathbf{v}_i = \sqrt{\lambda_i}\boldsymbol{\psi}_i$

iii) The collection $\{\boldsymbol{\psi}_i\}_{i=1}^r$ form an orthonormal basis for $R(\boldsymbol{\mathcal{X}})$.

Proof. This proof is a direct consequence of Theorem 2.7. □

The following corollary formalizes the eigentriples of $\boldsymbol{\mathcal{X}}$.

Corollary 4.7. *The collection of eigentriples, $\{\sqrt{\lambda_i}, \mathbf{v}_i, \boldsymbol{\psi}_i\}_{i=1}^r$, defines an MFSVD of $\boldsymbol{\mathcal{X}}$.*

4.4 Generalizing Multivariate Singular Spectrum Analysis to Multivariate Functional Singular Spectrum Analysis

We note that a key step in extending different SSA-based approaches, is how to obtain the trajectory matrix (operator) in the embedding step (see e.g., Subsections 1.3.3 and 4.2.1). Despite the fact that in SSA, the trajectory matrix is a linear combination of the associated L-lagged vectors, this is not the case for MSSA.

In Section 4.2, we obtain MFSSA by generalizing FSSA, where we introduce the multivariate trajectory operator as a linear combination of L-lagged multivariate function vectors that correspond to an MFTS. Alternatively, we may mimic the approach of MSSA algorithms (HMSSA or VMSSA) and develop new trajectory operators that are not necessarily based on L-lagged function vectors. The following subsections extend HMSSA and VMSSA to obtain the functional versions respectively.

4.4.1 From Horizontal Multivariate Singular Spectrum Analysis to Horizontal Multivariate Functional Singular Spectrum Analysis

We may view the column vectors of matrix $\mathbf{X}^{(j)}$ in equation (8), $\mathbf{x}_k^{(j)}$'s, as the univariate L-lagged vectors for the j^{th} variable. We see that $\mathbf{X}^{(j)}$ is an operator from $\mathbb{R}^K \to \mathbb{R}^L$, which can be seen as

a linear combination of these L-lagged vectors:

$$\mathbf{X}^{(j)}\mathbf{a}^{(j)} = \sum_{k=1}^{K} a_k^{(j)}\mathbf{x}_k^{(j)}, \qquad \mathbf{a}^{(j)} = \left[a_1^{(j)}, \ldots, a_K^{(j)}\right]^\top \in \mathbb{R}^K.$$

In the embedding step of HMSSA, the trajectory matrix, given in equation (9), can be seen as a linear operator, $\mathbf{X} : \mathbb{R}^{pK} \to \mathbb{R}^L$, where

$$\mathbf{X}\mathbf{a} = \sum_{j=1}^{p}\sum_{k=1}^{K} a_k^{(j)}\mathbf{x}_k^{(j)}, \qquad \mathbf{a} = \left[\left(\mathbf{a}^{(1)}\right)^\top \cdots \left(\mathbf{a}^{(p)}\right)^\top\right]^\top \in \mathbb{R}^{pK}. \tag{46}$$

In order to make the extension to the function space, we need to assume that the L-lagged function vectors (not multivariate) in HMFSSA, denoted with $\boldsymbol{x}_k^{(j)}$, are in the space $\mathbb{F}_j^L$, for $j = 1, \ldots, p$. But the linear combination of $\boldsymbol{x}_k^{(j)}$'s are well-defined if and only if $\mathbb{F}_1^L = \cdots = \mathbb{F}_p^L$, or equivalently $T_1 = \cdots = T_p$. We shall call the extension of this special case as HMFSSA and we present it Section C.3 of the Appendices.

4.4.2 From Vertical Multivariate Singular Spectrum Analysis to Vertical Multivariate Functional Singular Spectrum Analysis

In the embedding step of VMSSA, the trajectory matrix, given in equation (10), can be seen as a linear operator, $\mathbf{X} : \mathbb{R}^K \to \mathbb{R}^{pL}$, with

$$\mathbf{X}\mathbf{a} = \sum_{k=1}^{K} a_j \underline{\mathbf{x}}_k, \qquad \mathbf{a} = [a_1, \ldots, a_K]^\top \in \mathbb{R}^K \quad \text{and } \underline{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k^{(1)} \\ \vdots \\ \mathbf{x}_k^{(p)} \end{bmatrix} \in \mathbb{R}^{pL}.$$

To develop VMFSSA, we extend this operator to the function space, i.e., $\underline{\mathbf{x}}_k$ should belong to a new unfolded Hilbert space, $\mathbb{H}^{p,L} = \underbrace{\mathbb{F}_1 \times \cdots \times \mathbb{F}_1}_{L \text{ times}} \times \ldots \times \underbrace{\mathbb{F}_p \times \cdots \times \mathbb{F}_p}_{L \text{ times}}$. Here, each length pL function vector, $\underline{\boldsymbol{x}} \in \mathbb{H}^{p,L}$, is denoted by $\underline{\boldsymbol{x}} = \left(x_1^{(1)}, \cdots, x_L^{(1)}, \ldots, x_1^{(p)}, \cdots, x_L^{(p)}\right)$. It is easy to see that $\mathbb{H}^{p,L}$ is a Hilbert space equipped with inner product

$$\left\langle \underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \right\rangle_{\mathbb{H}^{p,L}} = \sum_{i=1}^{L}\sum_{j=1}^{p} \left\langle x_i^{(j)}, y_i^{(j)} \right\rangle_{\mathbb{F}_j} = \sum_{i=1}^{L} \langle \vec{x}_i, \vec{y}_i \rangle_{\mathbb{H}} = \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{H}^L} .$$

Therefore, there exists an isomorphic map, $\boldsymbol{\mathcal{U}} : \mathbb{H}^L \to \mathbb{H}^{p,L}$, defined by $\boldsymbol{\mathcal{U}}(\boldsymbol{x}) = \underline{\boldsymbol{x}}$, where $\boldsymbol{x} \in \mathbb{H}^L$ is a length L multivariate function vector. Now one may define the linear operator $\underline{\boldsymbol{\mathcal{X}}} : \mathbb{R}^K \to \mathbb{H}^{p,L}$,

specified with $\underline{\boldsymbol{x}}_k$'s, as

$$\underline{\boldsymbol{\mathcal{X}}}\mathbf{a} = \sum_{k=1}^{K} a_j \underline{\boldsymbol{x}}_k, \qquad \mathbf{a} \in \mathbb{R}^K \quad \text{and} \quad \underline{\boldsymbol{x}}_k \in \mathbb{H}^{p,L}.$$

The following theorem illustrates the equivalency between the results of the decomposition process applied in MFSSA and VMFSSA.

Theorem 4.8. *Let $\{\sqrt{\lambda_i}, \mathbf{v}_i, \boldsymbol{\psi}_i\}$ be the i^{th} eigentriple of $\boldsymbol{\mathcal{X}}$. The following holds:*

i) $\underline{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{X}}$.

ii) $\underline{\boldsymbol{\mathcal{X}}}$ *is a rank r operator with the i^{th} eigentriple denoted by $\{\sqrt{\lambda_i}, \mathbf{v}_i, \underline{\boldsymbol{\psi}}_i\}$, where $\underline{\boldsymbol{\psi}}_i = \boldsymbol{\mathcal{U}}\boldsymbol{\psi}_i$.*

Proof. We prove both parts of this theorem in the following.

i) Let $\mathbf{a} \in \mathbb{R}^K$. Then we have that

$$\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{X}}(\mathbf{a}) = \sum_{k=1}^{K} a_k \boldsymbol{\mathcal{U}}\boldsymbol{x}_k = \sum_{k=1}^{K} a_k \underline{\boldsymbol{x}}_k = \underline{\boldsymbol{\mathcal{X}}}(\mathbf{a})$$

and as such, we have that $\underline{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{X}}$.

ii) Again, let $\mathbf{a} \in \mathbb{R}^K$, then we have

$$\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{X}}(\mathbf{a}) = \sum_{i=1}^{r} \sqrt{\lambda_i}\mathbf{a}^\top \mathbf{v}_i \boldsymbol{\mathcal{U}}\boldsymbol{\psi}_i = \sum_{i=1}^{r} \sqrt{\lambda_i}\mathbf{a}^\top \mathbf{v}_i \underline{\boldsymbol{\psi}}_i = \underline{\boldsymbol{\mathcal{X}}}(\mathbf{a}).$$

This implies that the i^{th} eigentriple of $\underline{\boldsymbol{\mathcal{X}}}$ is $\left(\sqrt{\lambda_i}, \mathbf{v}_i, \underline{\boldsymbol{\psi}}_i\right)$ and that $\underline{\boldsymbol{\psi}}_i = \boldsymbol{\mathcal{U}}\boldsymbol{\psi}_i$.

□

Therefore the decompositions obtained via MFSSA and VMFSSA are interchangeable and subsequently the respective groupings and reconstructions are equivalent.

4.5 Numerical Studies

In order to explore the capabilities of MFSSA and HMFSSA, we implement a simulation study where we compare our two novel algorithms to other approaches of signal extraction for MFTS data. We

also present an application to remote sensing data which is used to further illustrate the interesting qualities of MFTS data that are discovered by MFSSA.

4.5.1 Simulation Study

For the simulation, we generate a bivariate FTS of lengths $N = \{100, 200\}$ by projecting the following discrete observations, sampled over regular intervals in i and t, onto a B-spline basis with 15 basis elements

$$\begin{aligned} Y_t^{(1)}(s_i) &= y_t^{(1)} + X_t^{(1)} \\ Y_t^{(2)}(s_i) &= y_t^{(2)} + X_t^{(2)}, \quad s_i \in [0,1], \quad i = 1, \ldots, 100, \quad t = 1, \ldots, N. \end{aligned}$$

We have that $y_t^{(1)} = \mu_t + \delta_t^{(1)}$ and $y_t^{(2)} = \delta_t^{(2)}$ are nonrandom, true signal terms. We take $\mu_t = \kappa t$ as an increasing trend component with $\kappa = \{0.00, 0.02\}$, $\delta_t^{(j)}$ are taken as seasonal components with expressions given as

$$\begin{aligned} \delta_t^{(1)} &= e^{s_i^2}\cos(2\pi\omega_1 t) - e^{1-s_i^2}\cos(2\pi\omega_2 t) - \sin(2\pi\omega_1 t)\cos(4\pi s_i) \\ &\quad + \sin(2\pi\omega_2 t)\sin(\pi s_i) \\ \delta_t^{(2)} &= e^{s_i^2}\sin(2\pi\omega_1 t) + \cos(2\pi\omega_1 t)\cos(4\pi s_i), \end{aligned}$$

where $\omega_1 = \{0.1, 0.5\}$, $\omega_2 = \{0, 0.25\}$, and $X_t^{(j)}$ are error terms for $j = 1, 2$. We also have that $X_t(s)$ is a stochastic term that follows a functional autoregressive model of order 1, FAR(1), defined by equation (29), equation (30), and equation (31). In this case, We choose γ_0 such that the norm of the operator in equation (31) takes on values of 0, 0.5, and 0.9. In addition, the functions, $\varepsilon_t(s)$, are considered to be independent trajectories of standard Brownian motion over the unit interval. Due to the presence of a trend component and two frequencies, we require five components in all algorithms to reconstruct the true structures which is due to the fact that each of the two frequencies is expressed in a sine and a cosine term. We compare reconstruction results of MFSSA to the results of HMFSSA where each algorithm is ran on the simulated bivariate FTS. We also compare the MFSSA and HMFSSA results to that of FSSA, MSSA, and DFPCA ran on each simulated covariate independently of one another. For MSSA we specify that the $200 \times N$ data matrix, $\mathbf{Q}$, follows the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix},$$

such that each $100 \times N$ matrix, $\mathbf{Q}_j$, follows the form $\mathbf{Q}_j = \left[Y_t^{(j)}(s_i)\right]_{i=1,\ldots,100}^{t=1,\ldots,N}$ for $j = 1, 2$. For all of the SSA-based algorithms we set $L = \{20, 40\}$ and for all algorithms, we measure the error of each reconstruction with the following root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N \times n \times p} \sum_{j=1}^{p} \sum_{t=1}^{N} \sum_{i=1}^{n} \left(y_t^{(j)}(s_i) - \hat{y}_t^{(j)}(s_i)\right)^2},$$

where $\hat{y}_t^{(j)}(s_i)$ is the reconstruction of covariate j, at time point t, evaluated at point s_i. For every unique combination of parameters and error terms, we repeat the simulation 100 times and report the mean of the RMSE's in the following plots whose vertical axes are taken over a log scale.

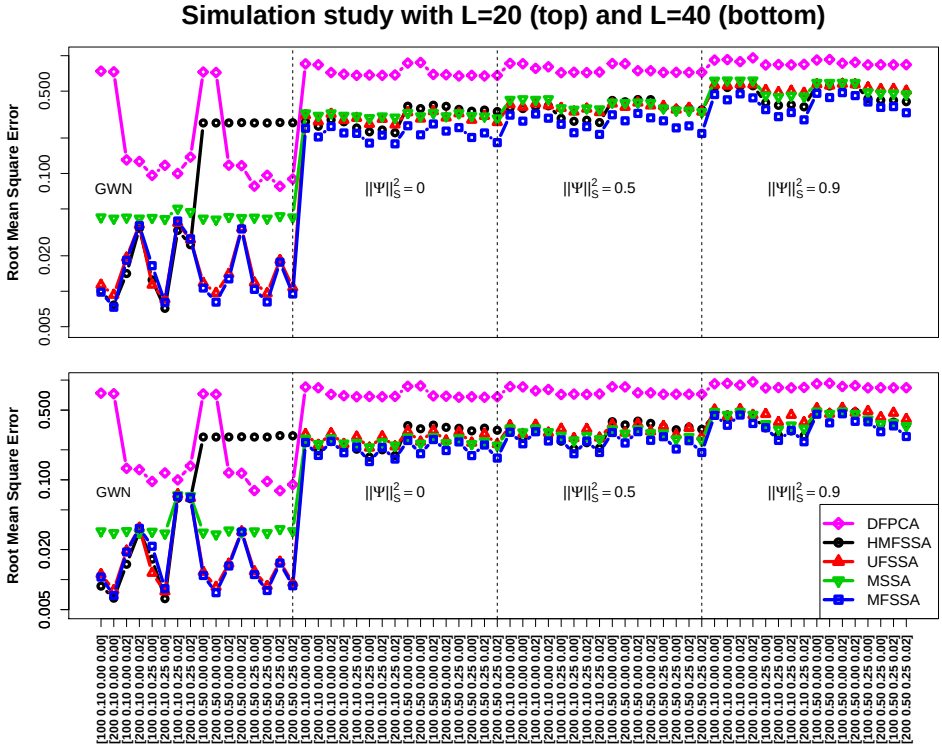

Figure 11: Simulation study where the vertical axes give outputs of average root mean square error for 100 replications of each simulation setup and each entry on the horizontal axis follows the form $[N, \omega_1, \omega_2, \kappa]$. The top plot is when $L = 20$ and the bottom plot is when $L = 40$.

We see in the top plot that $L = 20$ and in the bottom plot, $L = 40$, while the vertical lines separate out the simulated data by noise models and in addition, each tick mark on the horizontal should be read as $[N, \omega_1, \omega_2, \kappa]$. From these two subfigures, we find that MFSSA almost always outperforms other techniques of decomposition of a MFTS while HMFSSA also outperforms other techniques occasionally.

4.5.2 Application to Remote Sensing and Weather Station Data

Most researchers believe that the amount of vegetation present within a region is correlated to the region's temperature. Researchers can use this correlation to get a better understanding of how the vegetation and temperature in a region changes over time. Data that tracks the intraday hourly mean temperature, in celsius, for a variety of United States weather stations is available for download from Diamond et al. (2013). In addition, satellite images of varying resolutions, regions, time periods, spectral bands, and their variants have been made available for download and analyzed using various techniques (Tuck et al., 2014). The NDVI measure, recorded in the satellite images, is bounded between zero and one and is used to track the amount of vegetation present in the image (Lambin, 1999) and an in-depth discussion on how such NDVI images have been used in analysis can be found in Subsection 2.4.2.

It was determined that using MSSA over SSA can lead to richer analysis of correlated data (Golyandina et al., 2015). If a variable with strong seasonality components and another variable with strong mean components are included together in an MSSA analysis, we expect to find strong seasonality and mean component reflected in the singular values and singular vectors. Similarily, we expect MFSSA applied to correlated functional variables will strengthen signal extraction results and now we give a real data study illustrating the improved signal extraction results.

We introduce a bivariate example of intraday hourly mean temperature curves and NDVI images of a parallelogram shaped region just east of Glacier National Park in Montana, U.S.A. located between longitudes of $113.30°\text{W} - 113.56°\text{W}$ and latitudes of $48.71°\text{N} - 48.78°\text{N}$. The curves and images were recorded every 16 days starting January 1, 2008 and ending September 30, 2013. We start by applying FSSA with a lag of 45 (to capture yearly periodic behavior in temperature and vegetative processes) to the function curves and smoothed images separately from one another and we obtain the following plots of the singular vectors.

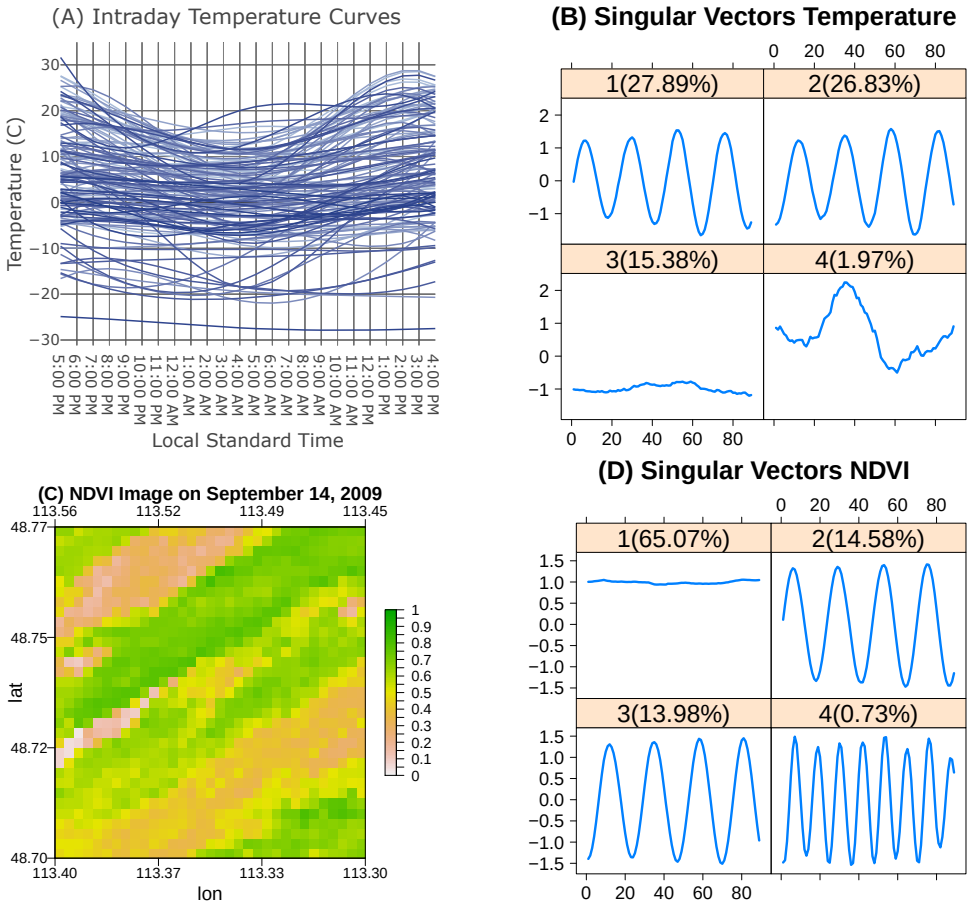

Figure 12: (A): Intraday temperature curves recorded in Saint Mary, Montana; (B): Right singular vectors from FSSA analysis applied to intraday temperature curves; (C): An NDVI image of Saint Mary, Montana; (D): Right singular vectors from FSSA analysis applied to NDVI images.

It is clear from Figure 12(B) that there exists a strong seasonality component in the intraday temperature curves of (A) accounting for 54.72% of the variation in the data while a mean behavior component accounts for 15.38% of the variation in the data. We also see from Figure 12(D) that the mean component captures 65.07% of the variation of the NDVI images data while the seasonality components only account for 28.56% of the variation of the data. We normalize the intraday temperature curves by dividing each sampling point by the standard deviation of all the sampling points (dividing by 10.13) since the NDVI images have values that are significantly smaller. We now apply MFSSA with a lag of 45 to the normalized intraday temperature curves and NDVI images in a bivariate analysis to obtain the following plots.

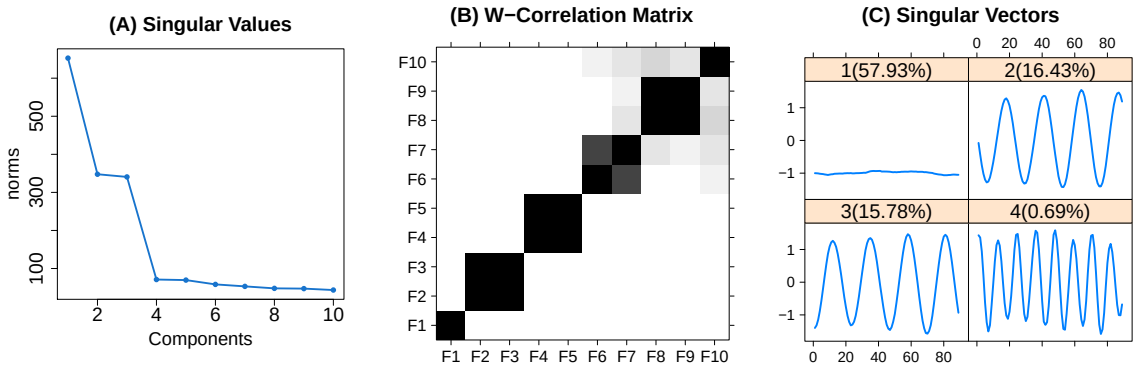

Figure 13: (A): Plot of singular values from MFSSA applied to intraday temperature curves and NDVI images from Saint Mary, Montana; (B): Plot of the w-correlation matrix generated from MFSSA; (C): Plot of right singular vectors from MFSSA.

Analyzing the plot of singular values and the w-correlation blocks in Figure 13(A) and (B) suggests that component one should be grouped by itself, components two and three should be grouped together, and four and five should be grouped together. Figure 13(C) shows that in the bivariate analysis, the mean component becomes dominant with the seasonal components taking on the second and third main sources of variation. This shows that combining the temperature curves and NDVI images functional data into a bivariate analysis reveals a stronger mean component as opposed to the weaker mean component seen in Figure 12(B). In addition to these presented results, we also offer visuanimations of the left singular functions in Section C.1 of the Appendices.

4.6 Discussion

Throughout this chapter, we presented MFSSA as a novel technique of dimension reduction of a MFTS. We found that the MFSSA problem is solved by performing VMFSSA and we also developed HMFSSA, presented in Section C.3 of the Appendices, as another approach but found that it was more restrictive and not as informative as MFSSA. We also developed MFSSA to be able to handle functions taken over different dimensional domains so that functional variables can take on many different forms. The MFSSA algorithm is available for use in the **Rfssa** R package (Haghbin et al., 2019), available through CRAN and in Section C.4 of the Appendices we give a description of a new S4 object, known as a **fts**, that is used to implement the different dimensional domains functionality of MFSSA in the code. We also offer another real data study using other forms of remote sensing data in Section C.2 of the Appendices.

CHAPTER 5: MULTIVARIATE FUNCTIONAL TIME SERIES FORECASTING: MULTIVARIATE FUNCTIONAL SINGULAR SPECTRUM ANALYSIS APPROACHES

5.1 Introduction

According to Chapter 4, if data are correlated, we may leverage this correlation to do a better job in performing signal extraction. We may also be interested in leveraging such correlation in nonparametric forecasts of MFTS data. In general, there has been little work done in forecasting MFTS data. Those existing approaches in nonparametric MFTS forecasting leverage functional principal component-based decompositions, see Gao and Shang (2017); Shang and Hyndman (2017) for more information. Given that our MFSSA algorithm offered less error as compared to other approaches in signal extraction of a simulated bivariate FTS as seen in Figure 11, it would be interesting to investigate whether this result can lead to more accurate SSA-based forecasts as compared with univariate FSSA for multivariate data. Our goal in the following chapter is to develop the MFSSA recurrent forecasting and MFSSA vector forecasting algorithms and to test whether or not we obtain better predictive accuracy from these novel methods as compared to their univariate counterparts applied to multivariate data. In Section 5.2 we derive the MFSSA recurrent and vector forecasting algorithms. Then in Section 5.3, we give the computer implementation strategies for both algorithms. In Section 5.4, we discuss some numerical results which show that for a MFTS of remote sensing data, the MFSSA forecasting approaches perform better in predictive accuracy as compared to their univariate counterparts. Finally, we finish with a discussion on results in Section 5.5.

5.2 Multivariate Functional Singular Spectrum Analysis Forecasting

Similar to Subsection 1.3.4, we are concerned with forecasting the deterministic nature of some MFTS, $\mathbf{y}_N$. As such, we apply the first two steps of the MFSSA routine to $\mathbf{y}_N$, as shown in Section 4.2, and now we develop the grouping and reconstruction stages in MFSSA for the purpose of forecasting. As an added note, we use the same notations and definitions given in Section 4.2 in the following to develop the forecasting routines.

MFSSA Grouping for Forecasting

Consider $\mathfrak{S}$ to be a subset of indices, $\{1, \ldots, r\}$, and define the operator, $\boldsymbol{\mathcal{X}}_{\mathfrak{s}} : \mathbb{R}^K \to \mathbb{H}^L$, given by $\boldsymbol{\mathcal{X}}_{\mathfrak{s}} = \sum_{i \in \mathfrak{S}} \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i$ where the definitions of $\mathbb{H}^L$, $\sqrt{\lambda_i}$, $\boldsymbol{\psi}_i$, $\mathbf{v}_i$, and $\mathbf{v}_i \otimes \boldsymbol{\psi}_i$ are presented in

Section 4.2. It is assumed that $\mathfrak{S}$ corresponds with a decomposition of the signal in $\mathbf{y}_N$. We find that we may express the multivariate trajectory operator, $\boldsymbol{\mathcal{X}}$, defined in Section 4.2, as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{\mathfrak{s}} + \boldsymbol{\mathcal{X}}_{\mathfrak{n}},$$

where the operator, $\boldsymbol{\mathcal{X}}_{\mathfrak{n}} = \sum_{i \in \mathfrak{S}^c} \sqrt{\lambda_i} \mathbf{v}_i \otimes \boldsymbol{\psi}_i$, represents the residual component, and $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ represents the signal component. Similar to equation (42), the operator, $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$, is specified by linear combinations of length L multivariate function vectors. We denote the length L multivariate function vectors that linearly span the range of $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ as $\boldsymbol{x}_j^{\mathfrak{s}} \in \mathbb{H}^L$ which are leveraged in MFSSA vector forecasting. Now in the reconstruction step, we extract an MFTS from $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ that is used in MFSSA recurrent forecasting.

MFSSA Reconstruction for Forecasting

We form the orthogonal projection of $\boldsymbol{\mathcal{X}}_{\mathfrak{s}}$ onto the space of Hankel operators that map from $\mathbb{R}^K$ to $\mathbb{H}^L$, which we denote with $\widetilde{\boldsymbol{\mathcal{X}}} \in \mathbb{H}_H^{L \times K}$, by calculating $\tilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{\Pi}_H \boldsymbol{\mathcal{X}}_{\mathfrak{s}}$, where $\mathbb{H}_H^{L \times K}$ and $\boldsymbol{\Pi}_H$ are defined in Chapter 4. From $\widetilde{\boldsymbol{\mathcal{X}}}$, we find that the reconstructed MFTS, $\tilde{\mathbf{y}}_N$, is given by $\tilde{\mathbf{y}}_N = \boldsymbol{\mathcal{T}}^{-1} \widetilde{\boldsymbol{\mathcal{X}}} = \left(\vec{\tilde{y}}_1, \ldots, \vec{\tilde{y}}_N\right)$ where $\boldsymbol{\mathcal{T}} : \mathbb{H}^N \to \mathbb{H}_H^{L \times K}$ is an invertible operator presented in Section 4.2.

Similar to the univariate FSSA case we find that the residual elements of $\mathbf{y}_N$ are captured in $\mathbf{y}_N - \tilde{\mathbf{y}}_N$ and we typically do not include these other modes of variation in the SSA-based forecasts. From here, we introduce our novel forecasting routines.

5.2.1 MFSSA Recurrent Forecasting and MFSSA Vector Forecasting

We begin with some notations that are used to construct our forecasting routines. For each length L multivariate function vector, $\boldsymbol{x} \in \mathbb{H}^L$, denote by $\boldsymbol{x}^{\nabla}$ and $\boldsymbol{x}^{\Delta}$ as the length $L-1$ multivariate function vectors consisting of the first and the last (respectively) $L-1$ components of $\boldsymbol{x}$. Setting $k < r$, let us define the function vector spaces, $\mathbb{L} = \text{span}\{\boldsymbol{\psi}_i\}_{i=1}^k$ and $\mathbb{L}^{\nabla} = \text{span}\{\boldsymbol{\psi}_i^{\nabla}\}_{i=1}^k$. We define the multivariate function vector, $\vec{\pi}_n$, to be the last component of the n^{th} left singular function, $\boldsymbol{\psi}_n$. From here, we define the operator, $\boldsymbol{\mathcal{V}} : \mathbb{H} \to \mathbb{H}$, given by, $\boldsymbol{\mathcal{V}} = \sum_{n=1}^k \vec{\pi}_n \otimes \vec{\pi}_n$, such that for some multivariate function vector, $\vec{f} \in \mathbb{H}$, we have

$$\boldsymbol{\mathcal{V}}\left(\vec{f}\right) = \sum_{n=1}^{k} \left\langle \vec{f}, \vec{\pi}_n \right\rangle_{\mathbb{H}} \vec{\pi}_n.$$

We define the set of length L multivariate function vectors, $\mathbb{E} = \{\left(\vec{0}, \ldots, \vec{0}, \vec{x}\right) | \vec{x} \in \mathbb{H} \text{ and } \vec{x} \neq \vec{0}\}$

and similar to Theorem 3.1, if the intersection of $\mathbb{E}$ with the set of the elements of $\mathbb{L}$, denoted with $\mathbb{E} \cap \mathbb{L}$, is equal to the empty set, then $\|\boldsymbol{\mathcal{V}}\| < 1$ where $\|\cdot\|$ is the operator norm for all operators that map from $\mathbb{H}$ to $\mathbb{H}$. Then just as in Corollary 3.2, we find that the operator, $(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} : \mathbb{H} \to \mathbb{H}$, exists and has the expression of

$$(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} = \sum_{l=0}^{\infty} \boldsymbol{\mathcal{V}}^l$$

where $\boldsymbol{\mathcal{I}} : \mathbb{H} \to \mathbb{H}$ is the identity operator. Now we define $\mathbf{g}_{N+M} = (\vec{g}_1, \ldots, \vec{g}_N, \vec{g}_{N+1}, \ldots, \vec{g}_{N+M}) \in \mathbb{H}^{N+M}$ as a MFTS of length $N+M$, where in the following algorithms (MFSSA recurrent and vector forecasting), the first N multivariate function vectors, $\{\vec{g}_i\}_{i=1}^{N}$, are close to $\tilde{\mathbf{y}}_N$, and the main goal is to predict the last M terms ($\vec{g}_{N+1}$ to $\vec{g}_M$).

MFSSA Recurrent Forecasting Algorithm

Similar to all types of SSA-based recurrent forecasting algorithms, we predict the next observation in a reconstructed MFTS with a linear combination of the previous $L-1$ terms. Just like in Theorem 3.3, if $\mathbb{E} \cap \mathbb{L} = \emptyset$, then for any length L multivariate function vector, $\boldsymbol{y} = (\vec{y}_1, \ldots, \vec{y}_L) \in \mathbb{L}$,we have that the last component may be expressed by

$$\vec{y}_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \vec{y}_j \tag{47}$$

where $\boldsymbol{\mathcal{A}}_j : \mathbb{H} \to \mathbb{H}$ is an operator defined by $\boldsymbol{\mathcal{A}}_j = \sum_{n=1}^{k} \vec{\psi}_{j,n} \otimes (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} \vec{\pi}_n$ and $\vec{\psi}_{j,n}$ is the j^{th} component of the n^{th} left singular function, $\boldsymbol{\psi}_n$. From here we define the following algorithm.

Algorithm A - MFSSA Recurrent Forecasting:

$$\vec{g}_i = \begin{cases} \vec{\tilde{y}}_i, & i = 1, \ldots, N \\ \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \vec{g}_{i+j-L}, & i = N+1, \ldots, N+M \end{cases} \tag{48}$$

and we call this method the *MFSSA R-Forecasting* algorithm for short. Now we move onto MFSSA vector forecasting.

MFSSA Vector Forecasting Algorithm

To begin, we give some notations. We define the operator $\boldsymbol{\mathcal{P}}^{\nabla} : \mathbb{R}^k \to \mathbb{H}^{L-1}$ and its adjoint in a similar manner as seen in the FSSA V-forecasting paragraph of Subsection 3.2.1 only now we specify that the left singular functions have components that are multivariate function vectors of the Hilbert space, $\mathbb{H}$. The goal of the MFSSA vector forecasting algorithm is to find a linear operator $\boldsymbol{\mathcal{Q}} : \mathbb{L} \to \mathbb{L}$ such that for some length L multivariate function vector, $\boldsymbol{y} \in \mathbb{H}^L$, given by, $\boldsymbol{y} = \boldsymbol{\mathcal{Q}}\boldsymbol{x}$, the distance between the length $L-1$ function vectors, $\boldsymbol{x}^{\Delta}$ and $\boldsymbol{y}^{\nabla}$, is minimal where $\boldsymbol{x} \in \mathbb{H}^L$ is a length L multivariate function vector. Similar to Proposition 3.4, we find that the orthogonal projection, $\boldsymbol{\Pi} : \mathbb{H}^{L-1} \to \mathbb{L}^{\nabla}$, defined by

$$\boldsymbol{\Pi} = \boldsymbol{\mathcal{P}}^{\nabla}\left(\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*}\boldsymbol{\mathcal{P}}^{\nabla}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{\nabla}\right)^{*},$$

can be used to obtain such a $\boldsymbol{y}^{\nabla}$. Now we use equation (47) and we obtain the last component of $\boldsymbol{y}$ (i.e., $\vec{y}_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \vec{y}_j^{\nabla}$). Similar to Proposition 3.5, we find that

$$\vec{y}_L = \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \vec{x}_j^{\Delta}.$$

Finally, we derive the linear operator, $\boldsymbol{\mathcal{Q}}$, as follows:

$$\boldsymbol{\mathcal{Q}}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{\Pi}\left(\boldsymbol{x}^{\Delta}\right) \\ \sum_{j=1}^{L-1} \boldsymbol{\mathcal{A}}_j \vec{x}_j^{\Delta} \end{pmatrix}, \qquad \boldsymbol{x} \in \mathbb{L}. \tag{49}$$

From here, we develop the following algorithm.

Algorithm B - MFSSA Vector Forecasting:

1. Define the length L multivariate function vectors given by

$$\boldsymbol{w}_j = \begin{cases} \boldsymbol{x}_j^{\mathfrak{S}}, & j = 1, \ldots, K \\ \boldsymbol{\mathcal{Q}}\boldsymbol{w}_{j-1}, & j = K+1, \ldots, K+M \end{cases} \tag{50}$$

2. Form the operator $\boldsymbol{\mathcal{W}} \in \mathbb{H}^{L \times (K+M)}$ whose range is linearly spanned by the set $\{\boldsymbol{w}_i\}_{i=1}^{K+M}$.

3. Hankelize $\boldsymbol{\mathcal{W}}$ in order to extract the MFTS $\mathbf{g}_{N+M}$.

4. The multivariate function vectors, $\vec{g}_{N+1}, \cdots, \vec{g}_{N+M}$, form the M terms of the MFSSA vector

forecast.

From hereafter, we refer to this algorithm as *MFSSA V-forecasting* for brevity. In the next Section, we present computer implementation of both algorithms.

5.3 Computer Implementation Strategy

We assume that $\{\vec{\nu}_i\}_{i=1}^{d}$ is a linearly independent basis for the function vector space, $\mathbb{H}_d$, which is a d-dimensional subspace of $\mathbb{H}$, where the expressions for $\vec{\nu}_i$, $\mathbb{H}_d$, and d can be found in Section 4.3. For any multivariate function vector, $\vec{f} \in \mathbb{H}_d$, there exists a unique d-dimensional vector, $\mathbf{c}_{\vec{f}} = \left[c_{\vec{f},1}, \ldots, c_{\vec{f},d}\right]^\top \in \mathbb{R}^d$, where $\vec{f} = \sum_{i=1}^{d} c_{\vec{f},i}\vec{\nu}_i$. We refer to $\mathbf{c}_{\vec{f}}$ as the corresponding vector of coefficients of $\vec{f}$. We also define $\mathbf{G} = \left[\langle \vec{\nu}_i, \vec{\nu}_j \rangle_{\mathbb{H}}\right]_{i,j=1}^{d}$ to be the $d \times d$ Gram matrix.

From hereafter, we consider all discussed operators whose domain or range are infinite dimensional Hilbert spaces to operate on or map to the corresponding d-dimensional subspace. For example, now we consider $\boldsymbol{\mathcal{V}}$ to be an operator that maps from $\mathbb{H}_d$ into $\mathbb{H}_d$. We now explore an implementation of $(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} : \mathbb{H}_d \to \mathbb{H}_d$ to be leveraged in the implementation of our algorithms. Similar to Lemma 3.6, we define $\mathbf{D} = \mathbf{G}^{\frac{1}{2}} \left[\mathbf{c}_{\vec{\pi}_1} \cdots \mathbf{c}_{\vec{\pi}_k}\right]$ to be a $d \times k$ matrix and we have that

$$\left[\left\langle (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{V}})^{-1} (\vec{\pi}_n), \vec{\nu}_i \right\rangle_{\mathbb{H}}\right]_{i=1,\ldots,d}^{n=1,\ldots,k} = \mathbf{G}^{\frac{1}{2}}\mathbf{D}\left(\sum_{l=0}^{\infty} \left(\mathbf{D}^\top\mathbf{D}\right)^l\right).$$

Remark: Note that if $\mathbb{E} \cap \mathbb{L} = \emptyset$, then $\|\boldsymbol{\mathcal{V}}\| < 1$ and the real-valued sequence, $\left(\|\boldsymbol{\mathcal{V}}^l\|\right)_{l\in\mathbb{N}}$, converges to zero monotonically. As a result, we truncate $\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l$ when $\|\left(\mathbf{D}^\top\mathbf{D}\right)^l\|_{Fr} \approx 0$ for some $l \in \mathbb{N}$ where $\|\cdot\|_{Fr}$ denotes the Frobenius norm of a $k \times k$ matrix.

5.3.1 MFSSA R-Forecasting Implementation

As according to equation (48), our goal is to find the matrices that implement the effect of each $\boldsymbol{\mathcal{A}}_j : \mathbb{H}_d \to \mathbb{H}_d$ for $j = 1, \ldots, L-1$. Similar to Theorem 3.7, if $\mathbb{E} \cap \mathbb{L} = \emptyset$, we find that the $d \times d$ matrix that corresponds with $\boldsymbol{\mathcal{A}}_j$ has the form

$$\mathbf{A}_j = \mathbf{D}\left(\sum_{l=0}^{\infty}\left(\mathbf{D}^\top\mathbf{D}\right)^l\right)\mathbf{E}_j^\top\mathbf{G}^{\frac{1}{2}}, \qquad \text{for } j = 1, \ldots, L-1,$$

where $\mathbf{E}_j = \mathbf{G}^{\frac{1}{2}}\left[\mathbf{c}_{\vec{\psi}_{j,1}} \cdots \mathbf{c}_{\vec{\psi}_{j,k}}\right]$ is a $d \times k$ matrix. The corresponding coefficients of the multivariate function vector $\boldsymbol{\mathcal{A}}_j\vec{f} \in \mathbb{H}_d$ are given by the d-dimensional vector, $\mathbf{A}_j\mathbf{c}_{\vec{f}} \in \mathbb{R}^d$. From here, we find that the corresponding coefficient vectors of the multivariate function vectors, $\vec{g}_i$'s, given in equation

(48), can be written as

$$\mathbf{c}_{\vec{g}_i} = \begin{cases} \mathbf{c}_{\vec{y}_i}, & i = 1, \ldots, N \\ \sum_{j=1}^{L-1} \mathbf{A}_j \mathbf{c}_{\vec{g}_{i+j-L}}, & i = N+1, \ldots, N+M \end{cases}. \tag{51}$$

5.3.2 MFSSA V-Forecasting Implementation

In order to obtain the implementation for the MFSSA V-forecasting algorithm, we first leverage theory developed in Section 4.3. For some positive integer, L, we define $\mathbb{H}_d^L$ to be the space created from the Cartesian product of L copies of $\mathbb{H}_d$. Now we define the quotient-remainder sequence

$$j = (q_j - 1)\, L + r_j, \quad 1 \leq q_j \leq d, \quad 1 \leq r_j \leq L,$$

and from here we define the collection of linearly independent length L multivariate function vectors, $\{\boldsymbol{\phi}_j\}_{j=1}^{Ld}$, where $\boldsymbol{\phi}_j \in \mathbb{H}_d^L$ is the zero function in all coordinates except for the r_j^{th} which is the multivariate function vector, $\vec{\nu}_{q_j} \in \mathbb{H}_d$. We find that the basis, $\{\boldsymbol{\phi}_j\}_{j=1}^{Ld}$, linearly spans $\mathbb{H}_d^L$. We find that any length L multivariate function vector, $\boldsymbol{f} \in \mathbb{H}_d^L$, can be represented via its corresponding Ld-dimensional coefficient vector $\mathbf{c}_{\boldsymbol{f}} = [c_{\boldsymbol{f},1}, \cdots, c_{\boldsymbol{f},Ld}]^\top \in \mathbb{R}^{Ld}$, where $\boldsymbol{f} = \sum_{i=1}^{Ld} c_{\boldsymbol{f},i} \boldsymbol{\phi}_i$. We denote the updated $Ld \times Ld$ Gram matrix with $\mathbf{H} = \left[\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle_{\mathbb{H}^L}\right]_{i,j=1}^{Ld}$. Now, we define the new truncated quotient-remainder sequence as

$$i = (q_i - 1)\,(L-1) + r_i, \quad 1 \leq q_i \leq d, \quad 1 \leq r_i \leq L-1.$$

This allows us to define $\{\boldsymbol{\phi}_i^\nabla\}_{i=1}^{(L-1)d}$ where $\boldsymbol{\phi}_i^\nabla \in \mathbb{H}_d^{L-1}$ is a length $L-1$ multivariate function vector which is zero in all coordinates except for the r_i^{th} which is $\vec{\nu}_{q_i}$, and $\mathbb{H}_d^{L-1}$ is the Hilbert space formed from the Cartesian product of $L-1$ copies of $\mathbb{H}_d$. Similar to the non-truncated case, the length $L-1$ multivariate function vectors, $\boldsymbol{f}^\nabla, \boldsymbol{f}^\Delta \in \mathbb{H}_d^{L-1}$, can be represented via the corresponding $(L-1)d$-dimensional coefficient vectors $\mathbf{c}_{\boldsymbol{f}^\nabla}, \mathbf{c}_{\boldsymbol{f}^\Delta} \in \mathbb{R}^{(L-1)d}$ associated to the basis functions $\{\boldsymbol{\phi}_i^\nabla\}_{i=1}^{(L-1)d}$. Also, we define the respective $(L-1)d \times (L-1)d$ Gram matrix, $\mathbf{H}^\nabla = \left[\left\langle \boldsymbol{\phi}_i^\nabla, \boldsymbol{\phi}_j^\nabla \right\rangle_{\mathbb{H}^{L-1}}\right]_{i,j=1}^{(L-1)d}$. Using these notations, we now give the impementation strategy for MFSSA V-forecasting algorithm.

To begin, we must find the matrix that implements $\boldsymbol{\Pi} : \mathbb{H}_d^{L-1} \to \mathbb{H}_d^{L-1}$. Similar to Theorem 3.9, we assume $\mathbb{E} \cap \mathbb{L} = \emptyset$ and we find that the $(L-1)d \times (L-1)d$ matrix that implements $\boldsymbol{\Pi}$ is given by

$$\mathbf{P} = \left(\mathbf{F}\mathbf{F}^\top + \mathbf{F}\mathbf{D}^\top \mathbf{D} \sum_{l=0}^{\infty} \left(\mathbf{D}^\top \mathbf{D} \right)^l \mathbf{F}^\top \right) \left(\mathbf{H}^\nabla \right)^{\frac{1}{2}},$$

where $\mathbf{F} = \left(\mathbf{H}^\nabla \right)^{\frac{1}{2}} \left[\mathbf{c}_{\boldsymbol{\psi}_1^\nabla} \cdots \mathbf{c}_{\boldsymbol{\psi}_k^\nabla} \right]$ is a $(L-1)d \times k$ matrix. In addition, we find that for any length $L-1$ multivariate function vector, $\boldsymbol{f} \in \mathbb{H}_d^{(L-1)}$, the corresponding coefficients of the length $L-1$ multivariate function vector, $\boldsymbol{\Pi f}$, are given by the $(L-1)d$-dimensional vector of coefficients, $\mathbf{P}\mathbf{c}_{\boldsymbol{f}}$. Now similar to Theorem 3.10, we let $\boldsymbol{x} \in \mathbb{L}$ be a length L multivariate function vector and we find that the corresponding coefficients of the length L multivariate function vector, $\boldsymbol{\mathcal{Q}x} \in \mathbb{L}$, are given by

$$\mathbf{c}_{\boldsymbol{\mathcal{Q}}}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{P}\mathbf{c}_{\boldsymbol{x}^\Delta} \\ \sum_{j=1}^{L-1} \mathbf{A}_j \mathbf{c}_{\vec{x}_j^\Delta} \end{bmatrix}.$$

Finally, we find that the corresponding coefficient vectors (real-valued vectors of length Ld) of the length L multivariate function vectors, $\boldsymbol{w}_j$'s, given in equation (50), can be written as

$$\mathbf{c}_{\boldsymbol{w}_j} = \begin{cases} \mathbf{c}_{\boldsymbol{x}_j^\mathfrak{s}}, & j = 1, \ldots, K \\ \mathbf{c}_{\boldsymbol{\mathcal{Q}}}(\boldsymbol{w}_{j-1}), & j = K+1, \ldots, K+M \end{cases}. \tag{52}$$

5.4 Numerical Study

In this Section, we show that for a bivariate FTS of remote sensing data, we obtain improved prediction results when applying MFSSA-based forecasting techniques as opposed to applying univariate FSSA forecasting methods on each covariate separately. We consider the NDVI density FTS seen in Figure 5(A) as our first variable and we derive the second variable from Enhanced Vegetative Index (EVI) remote sensing images. Like the NDVI data, EVI pixel values are bounded between 0 and 1 such that values closer to one are indicative of more vegetation and values closer to zero are indicative of less vegetation (Tuck et al., 2014). We note that EVI is an extension of NDVI in the sense that values are adjusted for atmospheric aerosol interference leading to a more conservative measurement of the amount of vegetation present in a region (Tuck et al., 2014). On the same days and in the same Jambi, Indonesia province that the NDVI images of plots (A) and (C) of Figure 4 were recorded, we obtain EVI images and from each image, we estimate a density of EVI as according to Silverman's rule of thumb Silverman (1986). From here, we obtain Figure 14 which

shows the bivariate FTS of $N = 448$ NDVI and EVI densities.

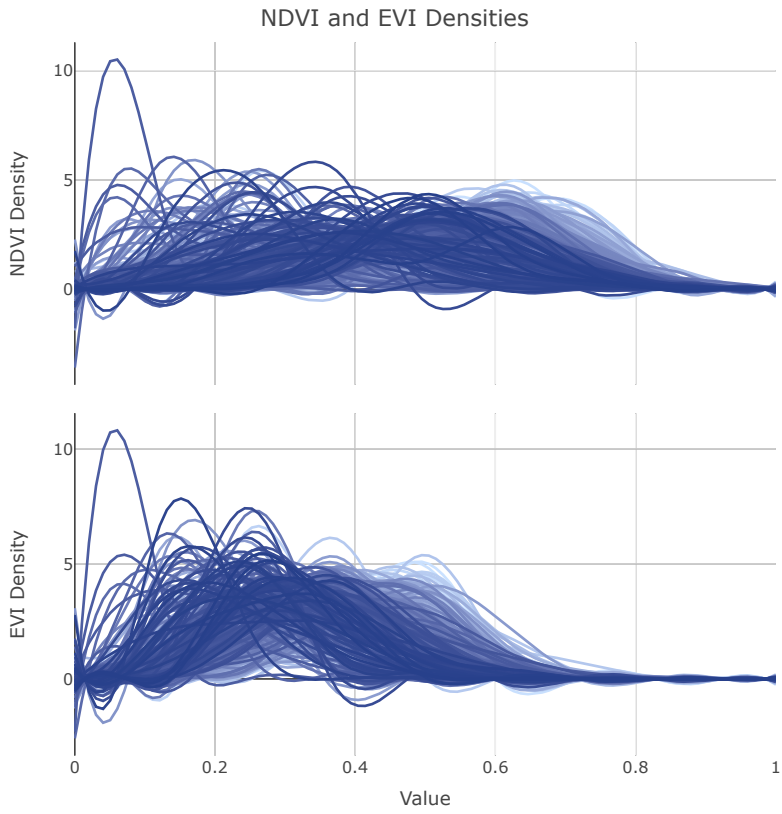

Figure 14: (top): The NDVI densities data; (bottom): The EVI densities data.

Our goal is to leverage a rolling forecast of the NDVI/EVI bivariate FTS using our MFSSA-based prediction methods and to also repeat this same process using univariate FSSA-based prediction techniques applied to the NDVI and EVI datasets independently of one another. We set the training set size to be $M = 403$ giving us a testing set size of 45 and we measure root mean square error of each method as according to

$$prRMSE = \sqrt{\frac{1}{(N-M) \times n \times p} \sum_{j=1}^{p} \sum_{t=1}^{N-M} \sum_{i=1}^{n} \left(y_t^{(j)}(s_i) - \hat{y}_t^{(j)}(s_i) \right)^2} \tag{53}$$

where $p = 2$, $n = 512$, $y_t^{(j)}(s_i)$ is observed function t of variable j evaluated at a point $s_i \in [0, 1]$ and $\hat{y}_t^{(j)}$ is the estimate of $y_t^{(j)}$. Table 4 gives the $prRMSE$'s, as according to equation (53), associated with each approach to MFSSA forecasting applied to the variables in a joint analysis and univariate FSSA forecasting methods applied to variables independently.

	MFSSA	FSSA
R-Forecasting	0.810	0.828
V-Forecasting	0.811	0.827

Table 4: $prRMSE$'s of MFSSA and FSSA MFTS forecasting techniques.

We see that our multivariate techniques outperform the univariate counterparts in predictive accuracy.

5.5 Discussion

In this chapter, we developed the forecasting techniques associated with MFSSA. We found that the methodologies perform better for the remote sensing NDVI/EVI data example as opposed to the univariate FSSA forecasting approaches. While this real data example is useful in showing that for the NDV/EVI data example, MFSSA forecasting methods should be used, it would be interesting to apply these developed methodologies to example data that utilizes the different dimensional domains flexibility seen in Chapter 4 which might lead to more accurate nonparametric forecasts of images. This is a topic of future work.

CHAPTER 6: CONCLUSION

The research goal of this dissertation was to establish mathematically and statistically rigorous algorithms that can perform nonparametric decomposition and forecasting of FTS data. In Chapter 2 we developed the novel FSSA approach which decomposes univariate FTS data and showed that the method can give more informative results as compared to other techniques of FTS signal extraction. Then in Chapter 3, we developed the new FSSA R-forecasting and FSSA V-forecasting methodologies that nonparametrically make predictions of FTS and we found that these techniques outperform current FTS forecasting gold standards for periodic stochastic processes. Next, we developed the MFSSA over different dimensional domains routine in Chapter 4 and we found that this technique can lead to improved accuracy in signal extraction as compared to univariate FSSA if data are correlated. Finally in Chapter 5, we finish by extending the MFSSA routine to perform nonparametric forecasting of a MFTS signal through MFSSA R-forecasting and MFSSA V-forecasting and found for the remote sensing data, the algorithms perform better in predictive accuracy as compared to the univariate counterparts. Throughout this work, the goal has been to use rigorous mathematical and statistical techniques to develop brand new algorithms that can lead to improved results and deeper scientific understanding of data. Given the results of this dissertation, we feel that this goal has been achieved and that the scientific field of functional data analysis has been advanced and the hope is that this work will inspire others to always reach for higher goals and to never stop improving.

BIBLIOGRAPHY

Alexandrov, T. (2009). A method of trend extraction using singular spectrum analysis. *RevStat*, 7(1):1–22.

Beyaztas, U. and Shang, H. L. (2019). Forecasting functional time series using weighted likelihood methodology. *Journal of Statistical Computation and Simulation*, 89(16):3046–3060.

Bosq, D. (2000). *Linear processes in function spaces: Theory and applications*. Number 149 in Lecture notes in statistics. Springer-Verlag, New York, NY USA.

Chiou, J.-M., Chen, Y.-T., and Yang, Y.-F. (2014). Multivariate functional principal component analysis: A normalization approach. *Statistica Sinica*, 24(4):1571–1596.

Diamond, H. J., Karl, T., Palecki, M. A., Baker, C. B., Bell, J. E., Leeper, R. D., Easterling, D. R., Lawrimore, J. H., Meyers, T. P., Helfert, M. R., Goodge, G., and Thorne, P. W. (2013). U.S. climate reference network after one decade of operations: status and assessment. https://www.ncdc.noaa.gov/crn/qcdatasets.html. Last accessed April 2020.

Fraiman, R., Justel, A., Liu, R., and Llop, P. (2014). Detecting trends in time series of functional data: A study of antarctic climate change. *Canadian Journal of Statistics*, 42(4):597–609.

Gao, Y. and Shang, H. L. (2017). Multivariate functional time series forecasting: Application to age-specific mortality rates. *Risks*, 5(2):21.

Golyandina, N., Korobeynikov, A., Shlemov, A., and Usevich, K. (2015). Multivariate and 2-D extensions of singular spectrum analysis with the Rssa package. *Journal of Statistical Software*, 67(2):1–78.

Golyandina, N., Korobeynikov, A., and Zhigljavsky, A. (2018). *Singular spectrum analysis with R*. Springer-Verlag, Berlin, Heidelberg DE.

Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. A. (2001). *Analysis of time series structure: SSA and related techniques*. Chapman and Hall/CRC, New York, NY USA.

Golyandina, N. and Osipov, E. (2007). The "Caterpillar"-SSA method for analysis of time series with missing values. *Journal of Statistical Planning and Inference*, 137(8):2642–2653.

Golyandina, N. and Zhigljavsky, A. (2013). *Singular spectrum analysis for time series*. Springer-Verlag, Berline, Heidelberg DE.

González, J. P., Muñoz San Roque, A. M. S., and Pérez, E. A. (2018). Forecasting functional time series with a new Hilbertian ARMAX model: Application to electricity price forecasting. *IEEE Transactions on Power Systems*, 33(1):545–556.

Grosjean, P. and Ibanez, F. (2018). *pastecs: Package for analysis of space-time ecological series*. R package version 1.3.21.

Groth, A. and Ghil, M. (2011). Multivariate singular spectrum analysis and the road to phase synchronization. *Physical Review E: Statistical, Nonlinear & Soft Matter Physics*, 84(3-2):036206–1 – 036206–10.

Haghbin, H., Morteza Najibi, S., Mahmoudvand, R., Trinka, J., and Maadooliat, M. (2021a). Functional singular spectrum analysis. *Stat.* e330 STAT-20-0240.R1.

Haghbin, H., Morteza Najibi, S., Mahmoudvand, R., Trinka, J., and Maadooliat, M. (2021b). Functional singular spectrum analysis supporting information. *Stat.* e330 STAT-20-0240.R1.

Haghbin, H., Najibi, S. M., Trinka, J., and Maadooliat, M. (2019). *Rfssa: Functional singular spectrum analysis*. R package version 1.0.0.

Happ, C. and Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522):649–659.

Hassani, H. and Mahmoudvand, R. (2013). Multivariate singular spectrum analysis: A general view and new vector forecasting approach. *International Journal of Energy and Statistics*, 1(1):55–83.

Hassani, H. and Mahmoudvand, R. (2018). *Singular spectrum analysis: Using R*. Palgrave Macmillan, London, UK.

Hassani, H., Rua, A., Silva, E. S., and Thomakos, D. (2019). Monthly forecasting of GDP with mixed-frequency multivariate singular spectrum analysis. *International Journal of Forecasting*, 35(4):1263 – 1272.

Hörmann, S., Kidziński, Ł., and Hallin, M. (2015). Dynamic functional principal components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(2):319–348.

Hörmann, S. and Kokoszka, P. (2012). *Functional time series*, volume 30 of *Handbook of Statistics*, pages 157–186. Elsevier B.V., Amsterdam, NL.

Hörmann, S., Kokoszka, P., Nisol, G., et al. (2018). Testing for periodicity in functional time series. *The Annals of Statistics*, 46(6A):2960–2984.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441.

Huang, J. Z., Shen, H., Buja, A., et al. (2008). Functional principal components analysis via penalized rank one approximation. *Electronic Journal of Statistics*, 2:678–695.

Hyndman, R. and Shang, H. L. (2009). Functional time series forecasting. *Journal of the Korean Statistical Society*, 38(3):199–211.

Hyndman, R. and Ullah, S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics & Data Analysis*, 51(10):4942–4956.

Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 31(3):300–303.

Kondrashov, D., Shprits, Y., and Ghil, M. (2010). Gap filling of solar wind data by singular spectrum analysis. *Geophysical Research Letters*, 37(15).

Lambin, E. F. (1999). Monitoring forest degradation in tropical regions by remote sensing: Some methodological issues. *Global Ecology and Biogeography*, 8(3-4):191–198.

Maadooliat, M., Huang, J. Z., and Hu, J. (2015). Integrating data transformation in principal components analysis. *Journal of Computational and Graphical Statistics*, 24(1):84–103.

Mahmoudvand, R. and Rodrigues, P. C. (2016). Missing value imputation in time series using singular spectrum analysis. *International Journal of Energy and Statistics*, 4(1):1650005.

Massy, W. F. (1965). Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60(309):234–256.

Mohammad, Y. and Nishida, T. (2011). On comparing SSA-based change point discovery algorithms. In *2011 IEEE/SICE International Symposium on System Integration (SII)*, pages 938–945. IEEE.

Moskvina, V. and Zhigljavsky, A. (2003). An algorithm based on singular spectrum analysis for change-point detection. *Communications in Statistics-Simulation and Computation*, 32(2):319–352.

Panuju, D. R. and Trisasongko, B. H. (2012). Seasonal pattern of vegetative cover from NDVI time-series. *Tropical Forests*, pages 255–268.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Prasetyo, L. B., Dharmawan, A. H., Nasdian, F. T., and Ramdhoni, S. (2016). Historical forest fire occurrence analysis in Jambi Province during the period of 2000–2015: Its distribution & land cover trajectories. *Procedia Environmental Sciences*, 33:450–459.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis.* Springer series in statistics. Springer-Verlag, New York, NY USA.

Ramsay, J. O. and Silverman, B. W. (2007). *Applied functional data analysis: Methods and case studies*. Springer-Verlag, New York, NY USA.

Reiss, P. T. and Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479):984–996.

Rodrigues, P. C. and Mahmoudvand, R. (2016). Correlation analysis in contaminated data by singular spectrum analysis. *Quality and Reliability Engineering International*, 32(6):2127–2137.

Shalit, O. M. (2017). *A first course in functional analysis.* Chapman and Hall/CRC, Boca Raton, FL USA.

Shang, H. L. (2013). Functional time series approach for forecasting very short-term electricity demand. *Journal of Applied Statistics*, 40(1):152–168.

Shang, H. L. (2019). A robust functional time series forecasting method. *Journal of Statistical Computation and Simulation*, 89(5):795–814.

Shang, H. L. and Hyndman, R. (2019). *rainbow: Bagplots, Boxplots and Rainbow Plots for Functional Data*. R package version 3.6.0.

Shang, H. L. and Hyndman, R. J. (2017). Grouped functional time series forecasting: An application to age-specific mortality rates. *Journal of Computational and Graphical Statistics*, 26(2):330–343.

Shen, H. and Huang, J. Z. (2005). Analysis of call centre arrival data using singular value decomposition. *Applied Stochastic Models in Business and Industry*, 21(3):251–263.

Shin, H., Yu, J., Jeong, Y., Wang, L., and Yang, D. (2017). Case-based regression models defining the relationships between moisture content and shortwave infrared reflectance of beach sands. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(10):4512 – 4521.

Silva, E. S., Hassani, H., and Heravi, S. (2018). Modeling european industrial production with multivariate singular spectrum analysis: A cross-industry analysis. *Journal of Forecasting*, 37(3):371 – 384.

Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman & Hall, London, UK.

Trefethen, L. N. and Bau III, D. (1997). *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA USA.

Trinka, J. (2019). Functional singular spectrum analysis and the clustering of time-dependent data. Master's thesis, Marquette University. `https://epublications.marquette.edu/mscs_stu/1`.

Trinka, J., Haghbin, H., and Maadooliat, M. (2020). Multivariate functional singular spectrum analysis over different dimensional domains. Version 1 retrieved from `https://arxiv.org/abs/2006.03933`.

Trinka, J., Haghbin, H., and Maadooliat, M. (2021). Functional time series forecasting: Functional singular spectrum analysis approaches. Version 4 retrieved from `https://arxiv.org/abs/2011.13077`.

Tuck, S. L., Phillips, H. R., Hintzen, R. E., Scharlemann, J. P., Purvis, A., and Hudson, L. N. (2014). MODISTools – downloading and processing MODIS remotely sensed data in R. *Ecology and Evolution*, 4(24):4658–4668.

University of California at Berkely (USA) and Max Planck Institue for Demographic Research (2020). Human mortality database. `https://www.mortality.org/`. Last accessed in August 2020.

Wagner-Muns, I. M., Guardiola, I. G., Samaranayke, V. A., and Kayani, W. I. (2018). A functional data analysis approach to traffic volume forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):878–888.

Weidmann, J. (1980). *Linear operators in Hilbert spaces*. Graduate texts in mathematics. Springer-Verlag, New York, NY USA.

APPENDICES

APPENDIX A: FUNCTIONAL SINGULAR SPECTRUM ANALYSIS SUPPLEMENT

This is the supplementary material for Chapter 2. Section A.1 includes a visuanimation and figures to help with visualization of the simulation study setups and results. Section A.2 gives an interesting application of FSSA to curves that desribe an anonymous bank's daily call center data where weekly patterns are uncovered. We note that another way to view the material presented here is through Haghbin et al. (2021b).

A.1 Further Illustrative Figures for the Simulation

The visuanimation of Figure A1 shows a sample of simulation setups where subplot (A) shows simulated functional data, subplot (B) shows the true signal, and subplots (C) and (D) show the results of FSSA and MSSA respectively for varying choices of the noise model and the frequency parameter (ω).

Figure A1: (A): The simulated signal and noise components added together; (B): The simulated true signal; (C): The signal extracted using FSSA; (D): The signal extracted using MSSA. Various simulation setups are generated by varying the noise structure and the frequency (ω) for fixed $L = 40$ and $N = 200$.

For further visualization, we include Figure A2 which depicts a perspective and a heatmap view of the FTS given by

$$m_t(s) = e^{s^2} \cos(2\pi\omega t) + \cos(4\pi s) \sin(2\pi\omega t), \quad s \in [0, 1], \text{ and } t = 1, \ldots, N,$$

where $N = 50$ and $\omega = 0.1$ and this is the form of the true signal given in Chapter 2, equation (28). We have that the plots of Figure A2 illustrate the periodicity that is present in the true signal when $\omega = 0.1$ due to the 10 unique types of curves that are generated.

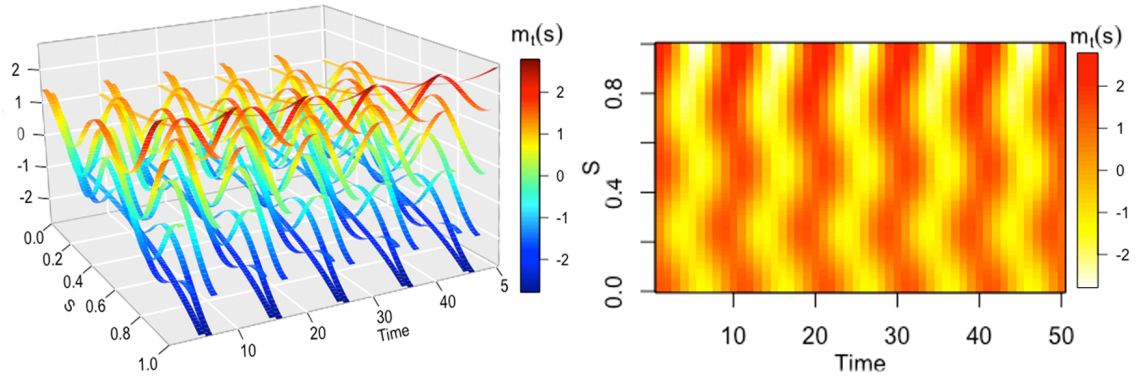

Figure A2: (left): A simulated true signal of 50 FTS observations with $\omega = 0.1$, viewed using a perspective plot; (right): A heatmap view of the same FTS shown in the perspective plot (left).

To illustrate the grouping of the first two eigentriples in the simulation study, we present several plots in Figure A3 using the **Rfssa** package. The plots of Figure A3 are the result of running FSSA with a lag of $L = 20$ on simulated FTS data where $N = 50$, $\omega = 0.1$, and $\|\Psi\|^2_{\mathcal{S}} = 0$. We also find that the plots of Figure A3 are the functional form (analogy) of the ones commonly used in the SSA literature (Golyandina et al., 2001). We see in the plot of the leading singular values (scree plot), the two leading singular values correspond to one group since the corresponding eigentriples (one and two) explain a similar amount of variation in the data. The w-correlation blocks confirm that the first two components should be assigned to one group. We use Figures A3(D) and A3(E) (right singular vectors and pairs of singular vectors), and we find that the eigentriple pairs, $1-2$, describe the periodicity in the FTS due to the oscillations in vector components and the fact that we obtain a decagon (see Golyandina et al. (2001) for more information) when we plot right singular vector one vs. right singular vector two in the pairs of singular vectors (1 vs. 2). Furthermore, some creative visualization tools that are implemented in the **Rfssa** package are used to extract within/between functional patterns for the simulated data by employing the estimated left singular functions and plots of these functions are given in Figures A3(C) and A3(F).

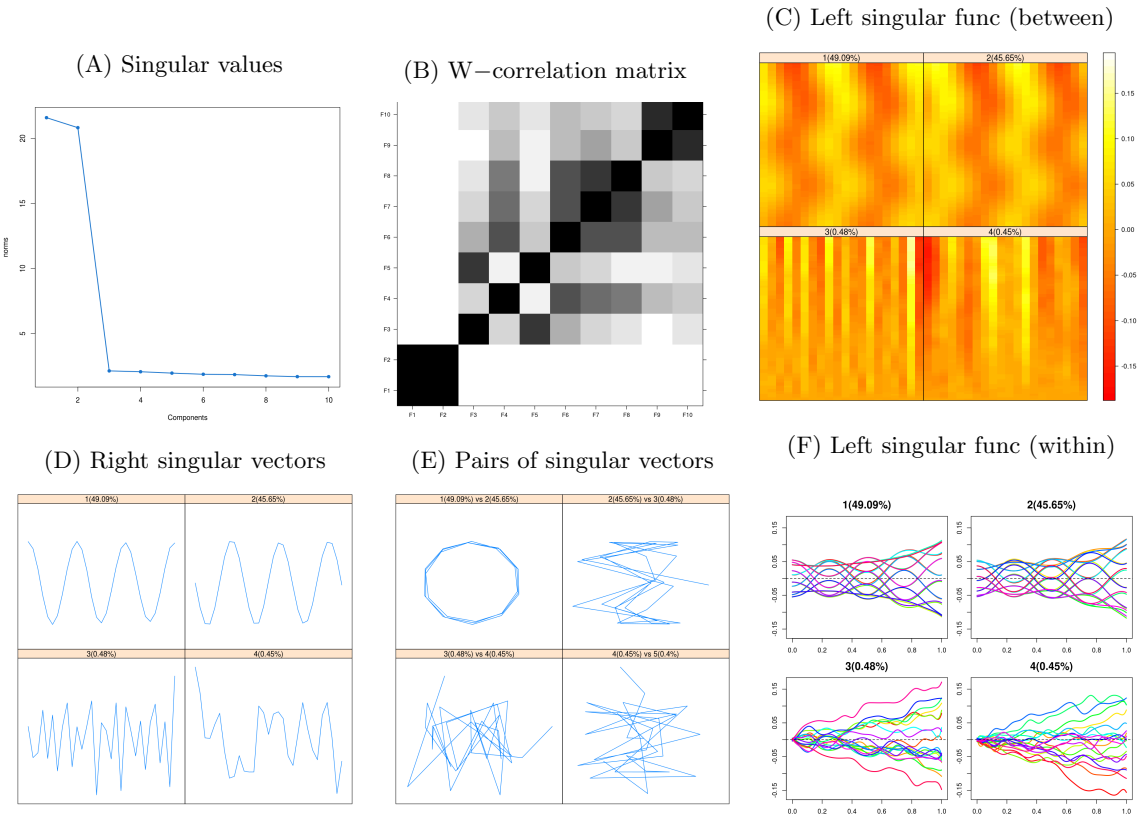

Figure A3: (A): Plot of singular values (scree plot); (B): Plot of the w-correlation matrix; (C): Heatmap of the first four left singular functions. (D): Plot of the first four right singular vectors; (E): Plot of pairs of right singular vectors; (F): Plot of the left singular functions.

A.2 Application to Call Center Dataset

To illustrate the advantages of FSSA, especially its main capability in extracting different functional components (i.e. trend, harmonic and noise), we explore the call center dataset analyzed in Maadooliat et al. (2015). This dataset provides the number of calls to a bank's call center, in Israel, per 6 minute intervals, between January 1 through December 31, 1999. Suppose $y_t(s_i),\ t = 1, \ldots, 365,\ i = 1, \ldots, 240$, is the square root of number of calls (to maintain shape but scale down observations) during the time interval s_i on day t. This call center data is analyzed in the motivating example of Chapter 2 and we show the observations again in Figure A4.

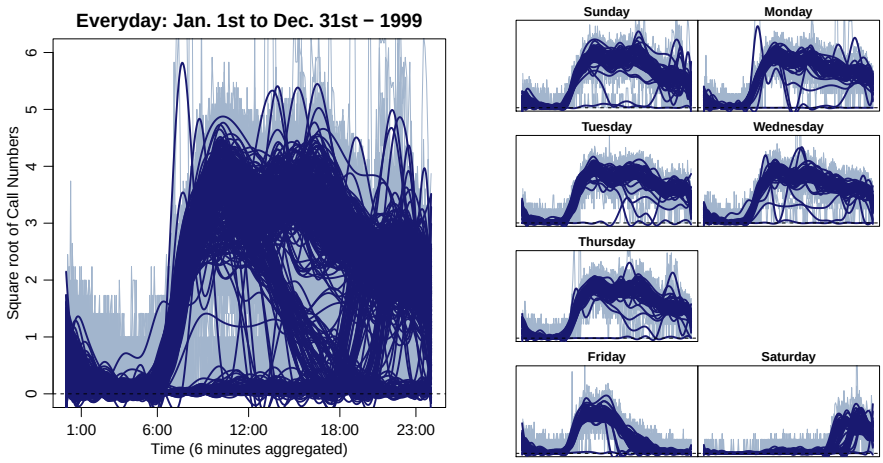

Figure A4: (left): The functional curves that give the number of calls to a bank's call center in Israel between January 1st to December 31st in the year 1999; (right): The same functional observations seen in the plot on the left, partitioned by weekday.

We have that Figure A4 shows the projection of the $y_t(s_i)$'s (vectors of length 240) onto a function space spanned by a cubic B-spline basis where the number of basis elements was determined using the GCV criterion.

An important goal of analyzing the call center data is to investigate the existence of periodic behaviors (e.g., weekly or monthly). Figure A4 visually confirms the existence of a strong weekly (work days vs. weekends) pattern in the dataset. Since one cannot visually distinguish different workday behaviors (Sunday through Thursday) or monthly behaviors using similar graphs, it would be interesting to show that FSSA can provide tools and machinery to separate out such signals.

In order to capture weekly patterns by FSSA, first, we choose the window length parameter, $L = 28$, (multiple of seven for the seven days of a week periodicity). Then we provide several plots using the **Rfssa** package for grouping the components (Figure A4). These plots are the functional form (analogy) of the ones commonly used in the SSA literature (Golyandina et al., 2001). As we see in the plot of leading singular values (scree plot), the first singular value is relatively large, and there exists three evident pairs of singular values where the singular values in each pair explain almost the same amounts of variability in the data. We assume that each of these three pairs correspond to a group. The w-correlation plot suggests partitioning the eigentriples into five groups: 1, $2-3$, $4-5$, $6-7$, $8-9$ and the remainder that does not seem to contain any strong signal. Considering the remaining plots (right singular vectors and pairs of right singular vectors), we see the eigentriple pairs $2-3$, $4-5$ and $6-7$ are related to a weekly periodicity with frequencies $1/7$, $2/7$ and $3/7$, while the last group, eigentriple pair $8-9$, describes a weak mode of variation that we find, in the

following, to be a weak monthly cycle.

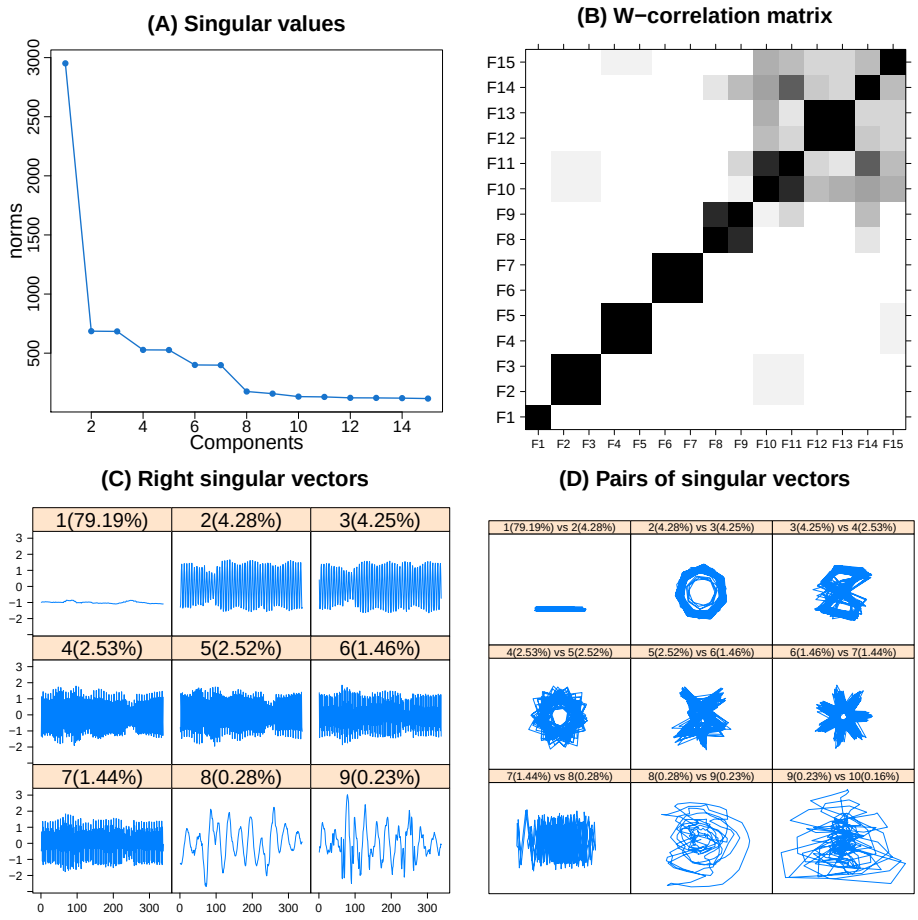

Figure A5: (A): The plot of singular values for FSSA with a lag of 28 applied to the call center data; (B): The plot of the w-correlation matrix; (C): The plot of the right singular vectors; (D): The plot of right singular vector pairs.

The groups are used reproduce the reconstructed FTS. The FTS associated with the first four groups are presented in Figure A6, which is drawn from the motivating example of Chapter 2.

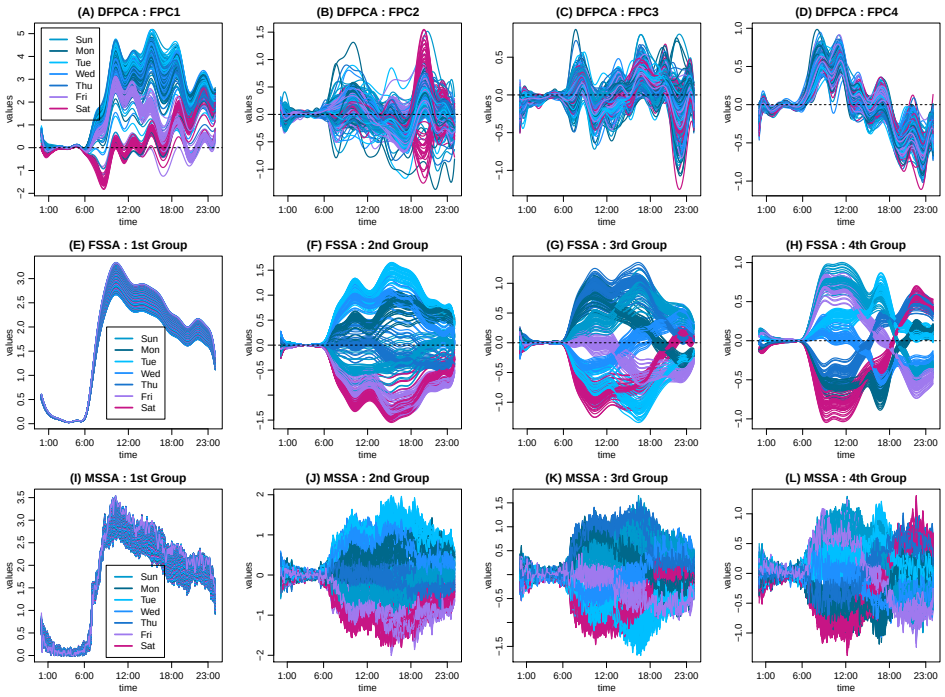

Figure A6: (top): The projection of the call center functional observations onto the four leading functional principal components (FPC)'s generated from DFPCA; (middle): The reconstructed FTS after grouping for FSSA; (bottom): The reconstructed multivariate time series and MSSA; Seven different colors are used to differentiate between different days of a week.

As mentioned in Chapter 2, using the FTS reconstructed using the fourth group in FSSA, we can visually distinguish seven different modes of variation that correspond with different days of the week including workdays. In short, we may use these results to cluster the call center FTS observations into belonging to one of the seven different days (separate workdays from one another) of the week and this analysis was the topic of the Master's essay of Trinka (2019). Furthermore, some creative visualization tools that are implemented in the **Rfssa** package can be used to extract within/between days patterns for the call center data by using the estimated left singular functions (Figure A5). We note that in Figure A5(B), there are 28 curves associated with each of the nine left singular functions (graphs). We also note that for the first left singular function, all curves resemble a similar pattern representative of the mean component. Left singular functions two through seven (inclusive) contain seven different patterns representative of the weekly periodicity in the data and each pattern consists of four curves whereas there's 28 distinct curves used to construct left singular functions eight and nine which might indicate the presence of monthly modes of variation (30 days). We explore the presence of a monthly mode of variation using a rigorous statistical test and we present the results in the following.

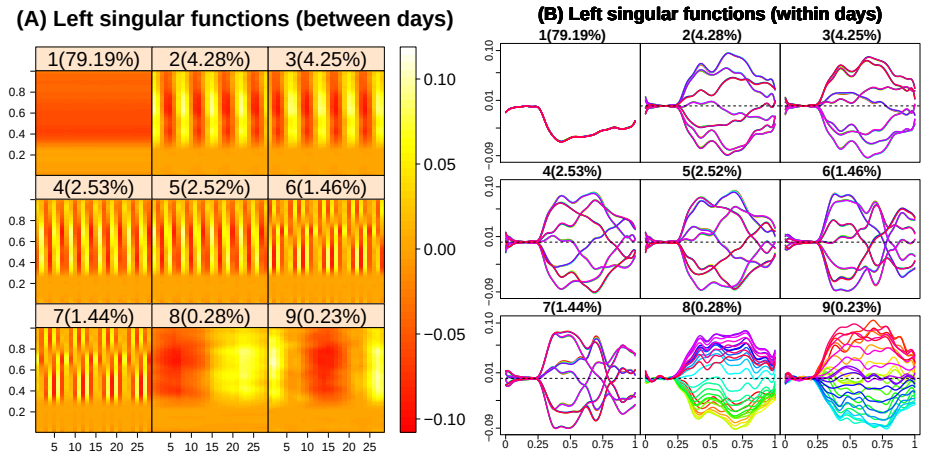

Figure A7: (A): Plot of left singular functions (between days); (B) Plot of left singular functions (within days).

We provide the multivariate trace periodicity test of Hörmann et al. (2018) on six sets of FTS (original signal $y_t(s)$, R_1, R_2, R_3, R_4, R_5), where R_i represents residual curves obtained via removing the reconstructed FTS generated by the first i groups, from the original signal $y_t(s)$. Table A1 provides the p-values of the test for the periods of length 7 and 30 days (p-values for testing the weekly and monthly patterns). It is clear that the periodicity test captures the weekly pattern for $y_t(s)$, R_1, R_2 and R_3 that contain either all or part of the weekly components (p-value=0). After subtracting the functional mean of all curves (first eigentriple) and the weekly components (eigentriples 2-7 inclusive), the monthly pattern in R_4 is not dominated anymore, and the periodicity test captures the monthly cycle in R_4 (p-value=0). This analysis confirms the existence of a weak monthly signal that is present in the call center FTS data. Finally, R_5 is the remainder of the signal after removing all of the weekly and monthly components, which is why the associated p-values in the last row are not significant anymore.

FTS	d=7	d=30
$y_t(s)$	0	0.86
R_1	0	0.96
R_2	0	0.97
R_3	0	0.30
R_4	1	0.00
R_5	1	0.17

Table A1: The p-values of the multivariate trace periodicity test of Hörmann et al. (2018) on six sets of FTS ($y_t(s)$, R_1, R_2, R_3, R_4, R_5) for the periods of length 7 and 30 days.

APPENDIX B: FUNCTIONAL SINGULAR SPECTRUM ANALYSIS FORECASTING SUPPLEMENT

The following contains supplementary materials to Chapter 3 in order to provide further clarification of the presented work. In Section B.1, we give a visuanimation which shows a subset of simulation setups and results from the simulation study of Subsection 3.4.1 for further visualization. Then in Section B.2, we give a review of the competing method of Hyndman and Ullah (2007). We note that another way to view the material presented here is through Trinka et al. (2021).

B.1 Further Illustrative Visuanimation for the Simulation

In order to further clarify what varying simulation setups and simulation results from Chapter 3 look like, we consider a subset of the different cases where we change the periodicity in the data, the trend, and the noise structure. We hold $L = 40$, $N = 200$, and $M = 80$ constant and we generate the visuanimation of Figure B1 which shows the true signal we are trying to predict in (A), the rolling forecast prediction of the testing set using the method of Hyndman and Ullah (2007) shown in (B), the rolling forecast prediction using FSSA R-forecasting in (C), and the rolling forecast prediction using FSSA V-forecasting in (D). The first four frames of the visuanimation are concerned with simulated data where only periodicity is present, the next four are concerned with cases where there is a mix of periodicity and trend, and the last four frames consider setups where there is only trend in the data.

Figure B1: (A): testing set true signal, (B): forecast of testing set using method of Hyndman and Ullah (2007), (C): FSSA R-forecast of testing set, (D): FSSA V-forecast of testing set. The last four frames in the visuanimation were generated using the **rainbow** package (Shang and Hyndman, 2019).

For the last four frames in the visuanimation, warmer colors are indicative of observations from earlier time points while cooler colors are indicative of observations of later time points.

B.2 Approach of Hyndman and Ullah to Functional Time Series Forecasting

The approach Hyndman and Ullah (2007) leverages a weighted functional principal component analysis of a FTS to find basis elements that explain variation. They then project their time dependent data onto the basis in order to find scores for each principal component. Finally, they perform forecasting techniques such as ARIMA on the resulting principal component scores in order to predict future observations of the FTS. The following steps outline the process.

1. Smooth the FTS using a nonparametric smoothing method to estimate $f_t(x)$ for $x \in [x_1, x_p]$

from sampling points $\{x_i, y_t(x_i)\}$, $i = 1, \ldots, p$ where $f_t(x)$ is a function vector observed on day t and p is the number of sampling points

2. Decompose the fitted observations using a weighted functional principal component analysis to obtain the basis expansion of $f_t(x)$ given by

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\phi_k(x) + e_t(x)$$

where $\mu(x)$ is the mean function, $\{\phi_k(x)\}_{k=1}^{K}$ is a set of orthonormal basis functions (notice we truncate at K basis elements), $e_t(x) \sim \mathrm{N}(0, \nu(x))$, and $\nu(x)$ is the covariance function

3. Fit univariate time series models, such as ARIMA, to each of the coefficients $\{\beta_{t,k}\}_{k=1}^{K}$

4. Forecast the coefficients $\{\beta_{t,k}\}_{k=1}^{K}$ for $t = n+1, \ldots, n+h$ using the fitted time series models where n is the length of the FTS and h is the forecast horizon

5. Obtain the function vector forecasts, $\{f_{n+1}(x), \ldots, f_{n+h}(x)\}$, by multiplying each $\beta_{t,k}$ by $\phi_k(x)$ and adding the resulting scaled function vectors together for $t = n+1, \ldots, n+h$

APPENDIX C: MULTIVARIATE FUNCTIONAL SINGULAR SPECTRUM ANALYSIS SUPPLEMENT

The following contains the supplementary material of Chapter 4. In Section C.1, we give visualizations of the left singular functions obtained in the MFSSA algorithm. In Section C.2 we give more applications of the MFSSA algorithm to remote sensing data. Then in Section C.3, we give full derivation of the HMFSSA algorithm as well as an application to remote sensing data. We finish with Section C.4 where we give the coding implementation strategy used to incorporate the different dimensional domains functionality into MFSSA. We note that another way to view the material presented here is through Trinka et al. (2020).

C.1 Left Singular Functions of Multivariate Functional Singular Spectrum Analysis

In this section, we build on the real data study presented in Chapter 4 by offering visualizations of the left singular functions that we found by applying FSSA and MFSSA to the intraday temperature data as well as the smoothed NDVI images (MFSSA is concerned with a joint analysis of the two variables). As mentioned in Chapter 4, we apply FSSA with a lag of 45 to the intraday temperature

curves and smoothed NDVI images separately. We also implement MFSSA with a lag of 45 to the temperature curves and NDVI images in a joint analysis. Figure C1 depicts the left singular functions corresponding to the intraday temperature curves that we obtain from the FSSA and MFSSA methods.

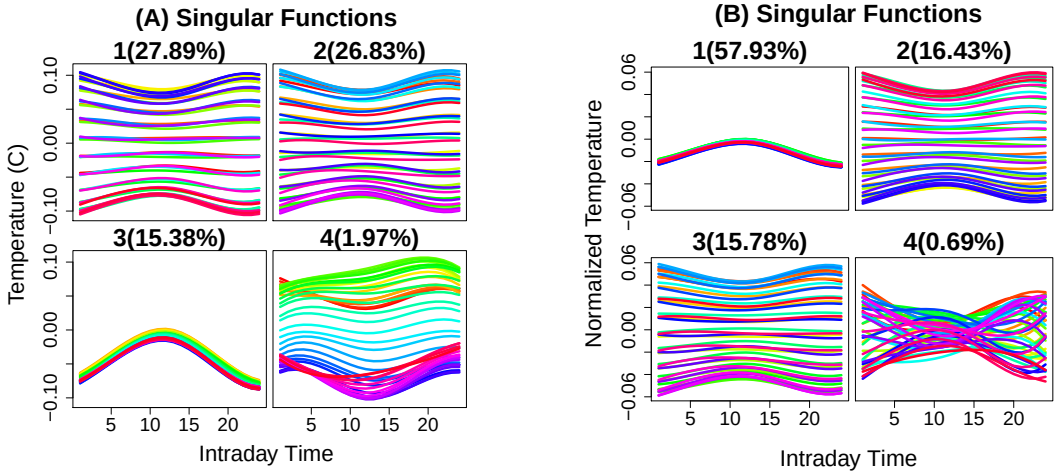

Figure C1: (A): All left singular functions that are found using FSSA with a lag of 45 applied to the intraday temperature curves; (B): All left singular functions that are found using MFSSA with a lag of 45 applied to the intraday temperature curves and smoothed NDVI images; (C): A visuanimation over time of the left singular functions presented in (A); (D): A visuanimation over time of the left singular functions presented in (B).

Figure C1(A) shows all $L = 45$ functions of the first four left singular functions of FSSA for the intraday temperature curves data while (C) steps through each function in a visuanimation. Figure C1(B) also shows all $L = 45$ functions of the first four left singular functions of MFSSA corresponding to the intraday temperature data while (D) steps through each function in a visuanimation. We see

that when MFSSA is applied to the intraday temperature curves and smoothed NDVI images, the mean component (first eigentriple) becomes stronger and explains more variability (57.93%) in the data as compared with the periodic components (eigentriples two and three). Now we investigate the left singular functions that correspond with the smoothed NDVI images.

Figure C2 depicts the resulting four left singular functions from FSSA and MFSSA in the following visuanimations.

Figure C2: (A): A visuanimation depicting the first four left singular functions that we obtain from FSSA applied with a lag of 45 to the smoothed NDVI images; (B): A visuanimation depicting the first four left singular functions that we obtain from MFSSA applied with a lag of 45 to the intraday temperature curves and smoothed NDVI images.

Here, we see little difference between the animations which indicates that the smoothed NDVI images are dominant in the analysis as compared with the intraday temperature curves.

C.2 Multivariate Functional Singular Spectrum Analysis Applied to Remote Sensing Density Curves

To further show that MFSSA enriches data analysis of correlated variables, we use a bivariate example of near-infrared (NIR) and shortwave infrared (SWIR) images taken every eight days of a region just outside of the city of Jambi, Indonesia between $103.61°\text{E} - 103.68°\text{E}$ and $1.67°\text{S} - 1.60°\text{S}$ over the timeline of February 18, 2000 and November 25, 2019. The wavelength of the NIR images ranges from 841-876 nanometers (nm) and the wavelength of the SWIR images are within the values of 2105-2155 nm. NIR light can be used for imaging vegetation as it is used in the calculation of the NDVI measure (Lambin, 1999) while shortwave infrared is often used for imaging the moisture

content in soil where a lower surface reflectance (SR) corresponds to higher moisture content (Shin et al., 2017). As mentioned in Prasetyo et al. (2016), it appears that this particular part of the Jambi province was a hot spot for controlled fires between 2001 and 2015 and the loss of vegetation over the course of about a decade will be reflected in lower NIR and higher SWIR SR values as time moves on. We obtain the KDEs of both the NIR and the SWIR SR images using Silverman's rule of thumb (Silverman, 1986) which we then project onto a cubic B-spline basis where the number of basis elements are chosen using the GCV criterion. In addition, we replaced outliers in the SWIR densities with the average of densities from the preceding and proceeding days. Similar results, as compared to the following, still hold even if the outliers are not removed. Applying FSSA with a lag of 45 to the NIR and SWIR densities separately, where this choice of lag approximately captures annual behavior, gives the following exploratory plots.

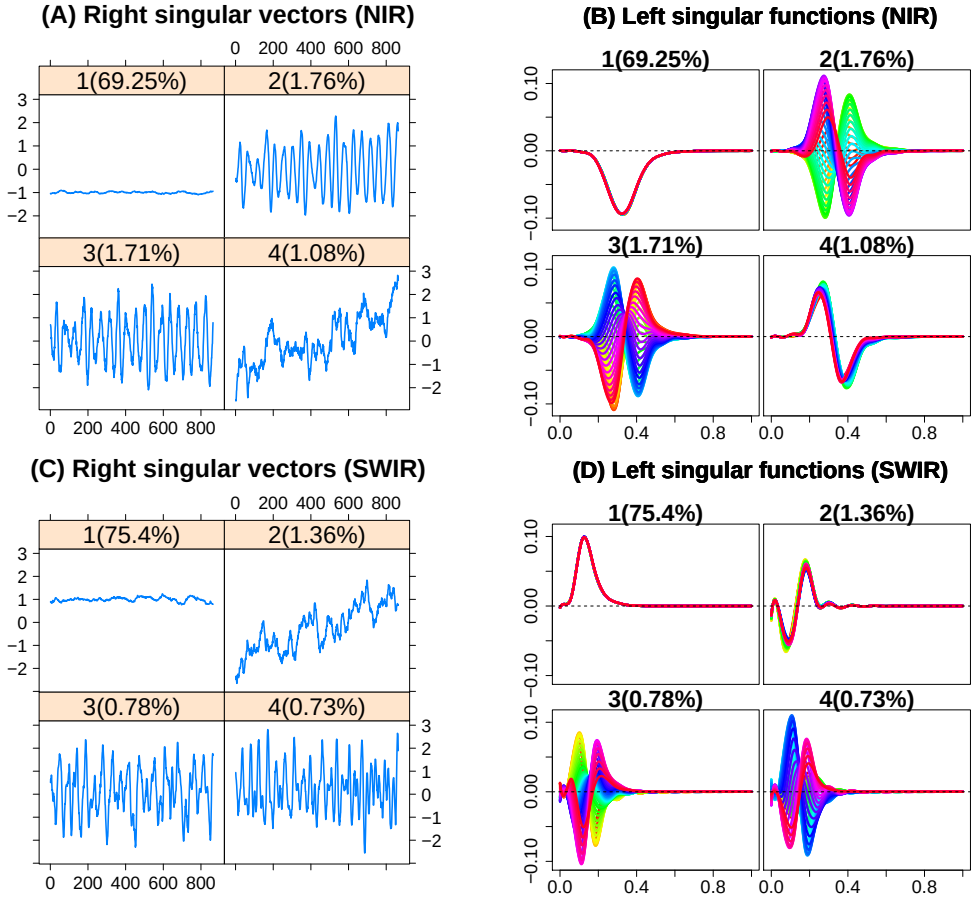

Figure C3: (A): The right singular vectors that are obtained when applying FSSA with a lag of $L = 45$ to the NIR densities data; (B): The left singular functions that are obtained when applying FSSA with a lag of $L = 45$ to the NIR densities data; (C): The right singular vectors that are obtained when applying FSSA with a lag of $L = 45$ to the SWIR densities data; (D): The left singular functions that are obtained when applying FSSA with a lag of $L = 45$ to the SWIR densities data.

Figure C3 subfigures (A) and (B) give us the right singular vectors and left singular functions of the NIR densities while Figure C3 subfigures (C) and (D) are the right singular vectors and left singular functions of the SWIR densities. We find that applying FSSA to the NIR densities captures seasonality in the second and third components while trend is present in the fourth component similar to the NDVI results of Chapter 2. Applying FSSA to the SWIR densities shows that trend is a more dominant behavior captured in the second component as compared to the seasonal behaviors captured in components three and four. Applying MFSSA decomposition with a lag of 45 to the bivariate NIR/SWIR example, where this lag is chosen to capture annual behavior, gives the following exploratory plots.

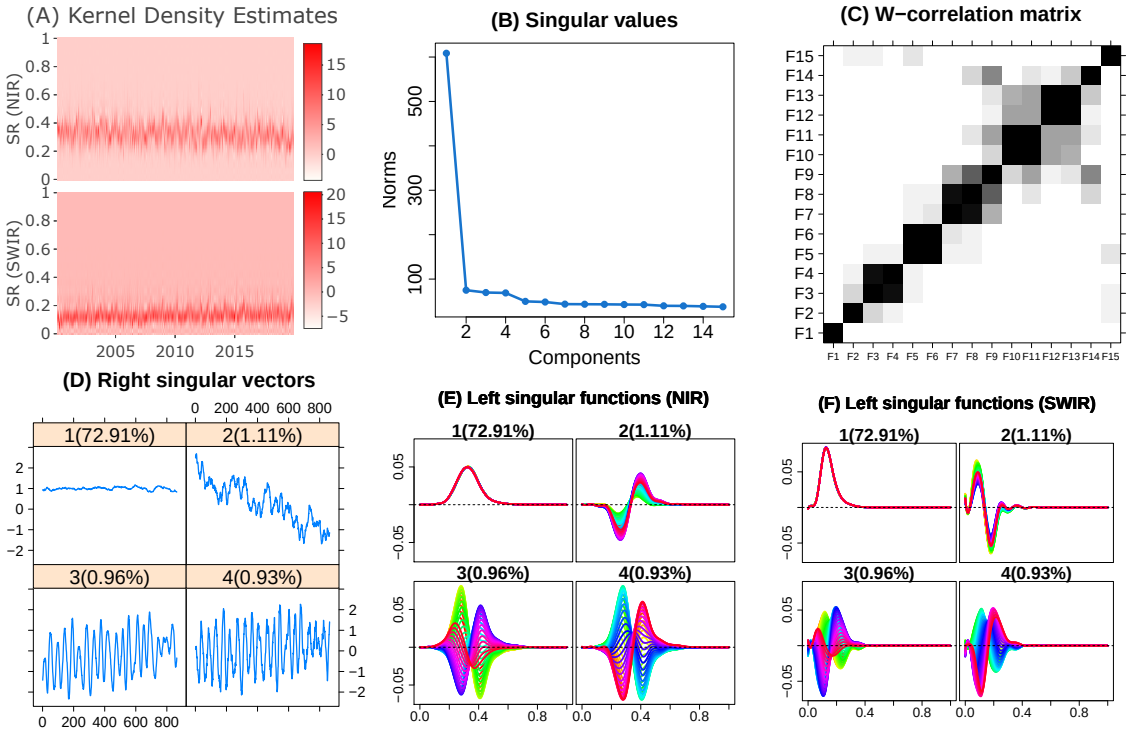

Figure C4: (A): Heatplots of the NIR and SWIR densities bivariate FTS; (B): The plot of the leading singular values that is generated from MFSSA applied with a lag of 45 to the bivariate FTS shown in (A); (C): The plot of the w-correlation matrix generated from the same MFSSA analysis that gave the singular values in (B); (D): The plot of the first four right singular vectors that was generated from the same MFSSA analysis that gave the singular values in (B); (E): The plot of the first four left singular functions that correspond with NIR densities that was generated from the same MFSSA analysis that gave the singular values in (B); (F): The plot of the first four left singular functions that correspond with SWIR densities that was generated from the same MFSSA analysis that gave the singular values in (B).

The bivariate FTS can be found in Figure C4 subfigure (A) while Figure C4 subfigures (B) and (C) are plots of singular values and w-correlation respectively. See that Figure C4 subfigure (D) gives us our MFSSA right singular vectors which showcases the weights that are multiplied by the left

singular functions shown in Figure C4 subfigures (E) and (F). Since we are performing MFSSA, we obtain 45 left eigenfunctions that correspond to the NIR densities as well as another set of 45 left eigenfunctions that correspond to the SWIR densities. Notice the trend behavior for the NIR densities is present in component two as according to Figure C4 subfigure (E) which indicates that adding SWIR densities into the analysis with the NIR densities created a more pronounced trend result as compared with Figure C3 subfigure (B). To this end, we find that performing a bivariate analysis on the NIR/SWIR densities enriched our data analysis as expected.

C.3 Horizontal Multivariate Functional Singular Spectrum Analysis

We begin this section with our discussion of moving from HMSSA to HMFSSA. As we clarified in Subsection 4.4.1 we need to assume that the domain $T = T_1 = \cdots = T_p$, $\mathbb{F} = L^2(T)$, and the L-lagged function vectors, $\boldsymbol{x}_k^{(j)}$'s, belong to a common Hilbert space, $\mathbb{F}^L$, which is formed from the Cartesian product of L copies of $\mathbb{F}$, for $j = 1, \cdots, p$. Notice that while the domain for each variable is the same, one may evaluate each variable at different points along T. We present the four main steps of the HMFSSA algorithm in the following subsection where we first form an operator whose range is built from L-lagged function vectors that describe MFTS behavior over sub-intervals of time.

Embedding, Decomposition, Grouping, and Reconstruction

We choose an integer L, such that $1 < L < \frac{N}{2}$, set $K = N - L + 1$, and we define the linear operator $\underset{\sim}{\boldsymbol{\mathcal{X}}} : \mathbb{R}^{pK} \to \mathbb{F}^L$ given by

$$\underset{\sim}{\boldsymbol{\mathcal{X}}}(\mathbf{a}) = \sum_{j=1}^{p} \sum_{k=1}^{K} a_k^{(j)} \boldsymbol{x}_k^{(j)}, \qquad \mathbf{a} \in \mathbb{R}^{pK}$$

which follows a similar form as compared to equation (46). The operator, $\underset{\sim}{\boldsymbol{\mathcal{X}}}$, is block Hankel, has rank $\tilde{r} \in \mathbb{N}$, where $1 \leq \tilde{r} \leq pK$, and we have that that the range of $\underset{\sim}{\boldsymbol{\mathcal{X}}}$ is given by $R\left(\underset{\sim}{\boldsymbol{\mathcal{X}}}\right) = \text{span}\{\boldsymbol{x}_k^{(j)}\}_{k=1,\ldots,K}^{j=1,\ldots,p}$. It is easy to see from the range of $\underset{\sim}{\boldsymbol{\mathcal{X}}}$ why all variables must share a common domain, T. Now we move to decomposition where we extract time-dependent modes of variation from $\underset{\sim}{\boldsymbol{\mathcal{X}}}$.

Since $\underset{\sim}{\boldsymbol{\mathcal{X}}}$ is a finite rank operator and thus compact, we utilize Theorem 7.6 from Weidmann (1980) to obtain an FSVD of $\underset{\sim}{\boldsymbol{\mathcal{X}}}$ given by

$$\underset{\sim}{\boldsymbol{\mathcal{X}}}(\mathbf{a}) = \sum_{i=1}^{\tilde{r}} \sqrt{\underset{\sim}{\lambda}_i} \left\langle \underset{\sim}{\mathbf{v}}_i, \mathbf{a} \right\rangle_{\mathbb{R}^{pK}} \underset{\sim}{\boldsymbol{\psi}}_i = \sum_{i=1}^{\tilde{r}} \sqrt{\underset{\sim}{\lambda}_i} \underset{\sim}{\mathbf{v}}_i \otimes \underset{\sim}{\boldsymbol{\psi}}_i(\mathbf{a}) = \sum_{i=1}^{\tilde{r}} \underset{\sim}{\boldsymbol{\mathcal{X}}}_i(\mathbf{a}),$$

where $\{\sqrt{\lambda_i}\}_{i=1}^{\tilde{r}}$ are the singular values, $\{\underset{\sim}{\mathbf{v}}_i\}_{i=1}^{\tilde{r}}$ are the orthonormal right singular vectors that span an $\tilde{r}$-dimensional subspace of $\mathbb{R}^{pK}$, and $\{\underset{\sim}{\boldsymbol{\psi}}_i\}_{i=1}^{\tilde{r}}$ are the orthonormal left singular functions that span an $\tilde{r}$-dimensional subspace of $\mathbb{F}^L$. We define $\{\sqrt{\lambda_i}, \underset{\sim}{\boldsymbol{\psi}}_i, \underset{\sim}{\mathbf{v}}_i\}$ to be the i^{th} eigentriple of $\underset{\sim}{\boldsymbol{\mathcal{X}}}$ and we group eigentriples together that describe similar MFTS behavior. Also notice that $\{\underset{\sim}{\boldsymbol{\mathcal{X}}}_i\}_{i=1}^{\tilde{r}}$ are rank one elementary operators built from the elements of the i^{th} eigentriple and we combine rank one operators within groups.

The grouping stage of HMFSSA is similar to the grouping stage of other types of SSA where we form a partition, $\{I_1, \ldots, I_m\}$, of the set of indices, $\{1, \ldots, \tilde{r}\}$, and we form operators, $\underset{\sim}{\boldsymbol{\mathcal{X}}}_{I_q} : \mathbb{R}^{pK} \to \mathbb{F}^L$, given by, $\underset{\sim}{\boldsymbol{\mathcal{X}}}_{I_q} = \sum_{i \in I_q} \underset{\sim}{\boldsymbol{\mathcal{X}}}_{I_q}$, for some positive integer q, where $1 \leq q \leq m$. In reconstruction we extract a MFTS from each $\underset{\sim}{\boldsymbol{\mathcal{X}}}_{I_q}$ that corresponds with a different mode of variation in the original MFTS, $\mathbf{y}_N$. We project each $\underset{\sim}{\boldsymbol{\mathcal{X}}}_{I_q}$ onto the subspace of block Hankel operators that map from $\mathbb{R}^{pK}$ to $\mathbb{F}^L$ to form a collection of m reconstructed MFTS, $\{\tilde{\mathbf{y}}_N^q\}_{q=1}^m$, where the projection is completed blockwise using the diagonal averaging technique of equation (22). Now we discuss the computer implementation of HMFSSA.

Computer Implementation Strategy

Computer implementation of HMFSSA is similar to that of Section 2.3 since the operator, $\underset{\sim}{\boldsymbol{\mathcal{X}}}$, maps to the Hilbert space, $\mathbb{F}^L$. Let $\{\nu_k\}_{k \in \mathbb{N}}$ be a known basis of the function space, $\mathbb{F}$, so that we may define the d-dimensional subspace $\mathbb{F}_d = \text{span}\{\nu_i\}_{i=1}^d$. Each function vector, $x \in \mathbb{F}_d$ can be uniquely represented by

$$x = \sum_{k=1}^{d} c_{k,i} \nu_k, \quad i = 1, \ldots, N, \quad c_{k,i} \in \mathbb{R}.$$

Let $\mathbb{F}_d^L$ be the d-dimensional subspace formed from the Cartesian product of L copies of $\mathbb{F}_d$, then the rest of the work in defining basis elements of $\mathbb{F}_d^L$ follows directly from Section 2.3. The work involving the expansion of the L-lagged function vectors, the range of $\underset{\sim}{\boldsymbol{\mathcal{X}}}$, the definition of the $Ld \times pK$ coefficient matrix. $\mathbf{B} = [b_{i,k}]_{i=1,\ldots,Ld}^{k=1,\ldots,pK} = [\mathbf{b}_1 \mathbf{b}_2 \cdots \mathbf{b}_{pK}]$, and the HMFSSA version of Theorem 4.6, also follows from Section 2.3 except we replace K with pK for the appropriate aforementioned quantities.

HMFSSA SWIR/NIR Study

In preliminary analysis, it was found that HMFSSA separates out MFTS behavior based on the covariate. In this subsection, we give an example of how HMFSSA attempts to separate out modes of variation within each variable without accounting for modes of variation in other variables (due

to similarities in ranges of the FSSA trajectory operator and HMFSSA trajectory operator). We apply HMFSSA with a lag of 45 to the NIR/SWIR example and obtain the following plots.

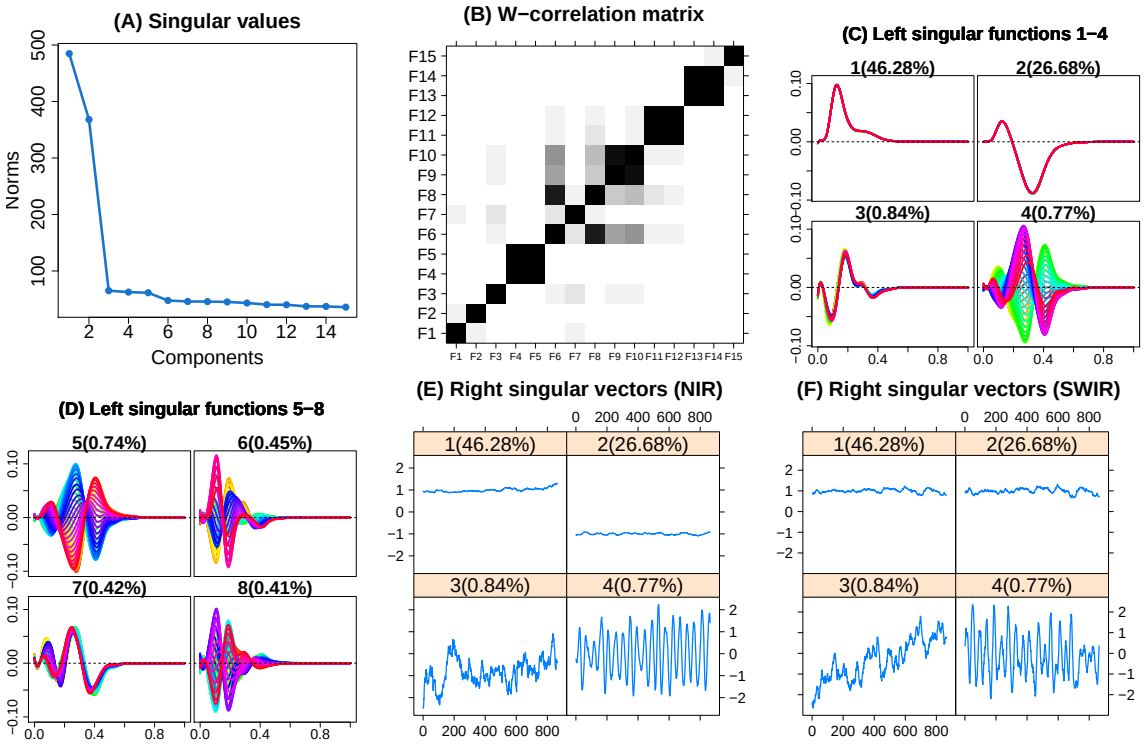

Figure C5: (A): The plot of singular values generated from applying HMFSSA to the NIR/SWIR bivariate FTS example with a lag of 45; (B): The plot of the w-correlation matrix generated from the same process used to generate (A); (C): The plot of the first four left singular functions generated from the same process used to generate (A); (D): The plot of singular functions five through eight (inclusive) generated from the same process used to generate (A); (E): The plot of the first four right singular vectors that correspond with the NIR densities generated from the same process used to generate (A); (F): The plot of the first four right singular vectors that correspond with the SWIR densities generated from the same process used to generate (A).

In this case, we have K right singular vectors that correspond to NIR densities and K right singular vectors that correspond to SWIR densities. It appears that the first component captures mean behavior of SWIR densities while the second component captures mean behavior of the NIR densities seen in Figure C5 subfigures (E) and (F) which is confirmed when we compare with Figure C3 subfigures (A) and (C). Rather than combining information to create a more pronounced mean component, HMFSSA works to separate out these behaviors by variable which is expected due to the similarity between HMFSSA and FSSA.

C.4 Rfssa fts Objects and Numerical Integration Methods

Currently, the **Rfssa** R package is heavily dependent upon the **fda** package. The goal is to migrate away from this dependency and rewrite the package in such a way that the user can create objects that are instances of the new S4 class named **fts**.

An object of the class **fts** will have the following attributes:

1. A list of length p where each entry contains a matrix of coefficients for the expansion of FTS observations for each of the p variables.
2. A list of length p containing grids over which each functional variable is defined.
3. A list of length p containing the basis matrices that were used to estimate the coefficients.

This migration will allow for greater flexibility as the user will not be restricted by basis elements defined only in the **fda** package and the **fssa()** function in the **Rfssa** package will become completely independent of the **inprod()** function of the **fda** package. In the following, we offer an example illustrating our proposed **fts** class with that of a bivariate FTS where one variable consists of curves and the other consists of surfaces taken over a two-dimensional domain. This same example is seen in Chapter 4 and the results of that portion of the dissertation were obtained using a local version of the **Rfssa** package that utilizes the proposed S4 **fts** object. In addition, we also offer our own numerical integration techniques used to estimate inner products between elements of Hilbert spaces in our **innerprod()** function.

Bivariate Functional Time Series

To begin we define a bivariate FTS. Let $T_1 = [a, b]$ and $T_2 = [c, d] \times [e, f]$ where $\times$ denotes the Cartesian product. Now we define $\mathbb{F}^{(j)} = L^2(T_j)$ as the Hilbert space of square integrable real functions defined on T_j for $j = 1, 2$. We define $\mathbb{H} = L^2(T_1) \times L^2(T_2)$ to be a Hilbert space where each multivariate function vector, $\vec{x} \in \mathbb{H}$, can be denoted by the 2-tuple $\left(x^{(1)}, x^{(2)}\right)$ for function vectors $x^{(1)} \in \mathbb{F}^{(1)}$ and $x^{(2)} \in \mathbb{F}^{(2)}$. Note that $\mathbb{H}$ is equipped with inner product

$$\langle \vec{x}, \vec{y} \rangle_{\mathbb{H}} = \sum_{j=1}^{2} \left\langle x^{(j)}, y^{(j)} \right\rangle_{\mathbb{F}^{(j)}} = \sum_{j=1}^{2} \int_{T_j} x^{(j)}(s_j)\, y^{(j)}(s_j)\, ds_j, \quad s_j \in T_j$$

where $\vec{x}, \vec{y} \in \mathbb{H}$. From here, we specify the bivariate FTS as $\mathbf{y}_N = (\vec{y}_1, \ldots, \vec{y}_N)$ where $\vec{y}_t \in \mathbb{H}$ for $t = 1, \ldots, N$. We are concerned with the representation of the bivariate FTS using our **fts**

object where we describe how functions over one-dimensional and two-dimensional domains will be represented. For clarity, we define the function vector, $x^{(1)} \in \mathbb{F}^{(1)}$, to be curve over a one-dimensional domain and each function vector $x^{(2)} \in \mathbb{F}^{(2)}$ to be a surface over a two-dimensional domain.

Functions Over a One-Dimensional Domain

Let $\{\nu_k^{(1)}\}_{k\in\mathbb{N}}$ denote a known linearly independent basis system of $\mathbb{F}^{(1)}$. Each function vector in $\mathbb{F}^{(1)}$ can be projected onto the d-dimensional subspace, $\mathbb{F}_d^{(1)}$, such that any function vector, $y_t^{(1)} \in \mathbb{F}_d^{(1)}$, can be expanded as

$$y_t^{(1)} = \sum_{k=1}^{d} \left\langle y_t^{(1)}, \tilde{\nu}_k^{(1)} \right\rangle_{\mathbb{F}^{(1)}} \nu_k^{(1)}$$

where $d_1 \in \mathbb{N}$ and the collection, $\{\tilde{\nu}_k^{(1)}\}_{k=1}^{d}$, is the dual basis of $\{\nu_k^{(1)}\}_{k=1}^{d}$. Now we discuss the techniques that are used to obtain $\left\langle y_t^{(1)}, \tilde{\nu}_k^{(1)} \right\rangle_{\mathbb{F}^{(1)}}$ to represent our functional observations.

Computer Representation

We represent functional data objects using matrices whose entries are the coefficients of each functional observation and this subsection is concerned with the estimation of such matrices using different numerical integration techniques. As a note, we consider all functional observations to be observed over a regular grid (discretization gives sampling points that are equally spaced from one another).

Consider the univariate FTS of size N, $\mathbf{y}_N^{(1)} = \left(y_1^{(1)}, \ldots, y_N^{(1)}\right)$, whose elements, $y_t^{(1)} \in \mathbb{F}^{(1)}$, are function vectors taken over the one-dimensional interval of $[a, b]$. Let $y_t^{(1)}(u_j)$ be a discrete sample of function $y_t^{(1)}$ at point $u_j \in [a, b]$ for $t = 1, \ldots, N$ and $j = 1, \ldots, M$. As such, we define the $M \times N$ matrix,

$$\mathbf{X} = \left[y_t^{(1)}(u_j)\right]_{j=1,\ldots M}^{t=1,\ldots,N}, \tag{C1}$$

to be the matrix that contains the discretized functional observations over the regular grid

$$u^* = \{u_1, \ldots, u_M\}. \tag{C2}$$

We also define the $M \times d$ matrix,

$$\mathbf{B} = \left[\nu_k^{(1)}(u_j)\right]_{j=1,\ldots,M}^{k=1,\ldots,d} \tag{C3}$$

to be the matrix that contains the discretized basis elements over the same regular grid, u^*.

Define $\Delta u = u_2 - u_1$ so that we obtain the estimate of the $d \times d$ Gram matrix, $\mathbf{G} = \left[\left\langle \nu_i^{(1)}, \nu_j^{(1)} \right\rangle_{\mathbb{F}^{(1)}}\right]_{i,j=1}^{d}$, via rectangular integration by computing

$$\mathbf{G} = \Delta u \mathbf{B}^\top \mathbf{B}. \tag{C4}$$

In addition, we obtain the $d \times N$ matrix, $\mathbf{C} = \left[\left\langle y_t^{(1)}, \tilde{\nu}_k^{(1)} \right\rangle_{\mathbb{F}^{(1)}}\right]_{k=1,\ldots,d}^{t=1,\ldots,N}$, via the least-squares solution with

$$\mathbf{C} = \mathbf{G}^{-1}\mathbf{B}^\top \mathbf{X}. \tag{C5}$$

In order to increase the accuracy of our estimate, we now offer the new estimation of $\mathbf{G}$ using the one-dimensional trapezoidal rule. We introduce the diagonal $M \times M$ matrix, $\mathbf{D}$, whose main diagonal is the vector

$$[1, 2, \ldots, 2, 1]^\top \in \mathbb{R}^M.$$

We may estimate $\mathbf{G}$ via the trapezoidal rule with

$$\mathbf{G} = \frac{\Delta u}{2}\mathbf{B}^\top \mathbf{D}\mathbf{B}.$$

Now we expand on these results for the representation of function vectors that are surfaces over a two-dimensional domain.

Functions over a Two-Dimensional Domain

Let $\{\alpha_i\}_{i\in\mathbb{N}}$ denote a linearly independent basis system of $L^2([c,d])$ and $\{\beta_j\}_{j\in\mathbb{N}}$ be a linearly independent basis system of $L^2([e,f])$. Now we define the d_1-dimensional and d_2-dimensional subspaces, $L^2_{d_1}([c,d]) \subset L^2([c,d])$ and $L^2_{d_2}([e,f]) \subset L^2([e,f])$, respectively. We define the function vector space $\mathbb{F}^{(2)}_d = L^2_{d_1}([c,d]) \times L^2_{d_2}([e,f])$ to be the d-dimensional subspace of $\mathbb{F}^{(2)}$ where $d = d_1 * d_2$. We com-

pute the linearly independent basis system of $\mathbb{F}_d^{(2)}$ as

$$\{\nu_k^{(2)}\}_{k=1}^{d} = \{\alpha_i \otimes \beta_j\}_{i=1,\ldots,d_1}^{j=1,\ldots,d_2}$$

where $k = (j-1)d_1 + i$ and $\otimes$ denotes the Kroenecker product. Using this basis, we find that we may expand any function vector, $y_i^{(2)} \in \mathbb{F}_d^{(2)}$, as

$$y_i^{(2)} = \sum_{k=1}^{d} \left\langle y_i^{(2)}, \tilde{\nu}_k^{(2)} \right\rangle_{\mathbb{F}^{(2)}} \nu_k^{(2)}.$$

where $\{\tilde{\nu}_k^{(2)}\}_{k=1}^{d}$ is the dual basis of $\{\nu_k^{(2)}\}_{k=1}^{d}$.

Computer Representation

Similar to the one-dimensional case, this Subsection is concerned with the estimation of matrices that are used to represent functional data objects. Since the observations are surfaces, grids will be taken to be regular and two-dimensional (discretization of the grid gives sampling points that are equally spaced along both dimensions of the grid).

Consider the univariate FTS of length N, $\mathbf{y}_N^{(2)} = \left(y_1^{(2)}, \ldots, y_N^{(2)}\right)$, where the function vector, $y_t^{(2)} \in \mathbb{F}_d^{(2)}$, is a surface taken over the two-dimensional domain of T_2. Let $u_k = \{v_i, w_j\} \in T_2$ where v_i is a point in $[c, d]$ for $i = 1, \ldots, M_1$, w_j is a point in $[e, f]$ for $j = 1, \ldots, M_2$, $M_1,\ M_2 \in \mathbb{N}$, and $k = (j-1)M_1 + i$. We define $M = M_1 \times M_2$ and we calculate the $M \times N$ matrix,

$$\mathbf{X} = \left[y_t^{(2)}(u_k)\right]_{k=1,\ldots,M}^{t=1,\ldots,N} \tag{C6}$$

to be the matrix that contains the discretized functional observations over the regular grid

$$u^* = \{u_1, \ldots, u_M\}. \tag{C7}$$

We also define the $M \times d$ matrix,

$$\mathbf{B} = \left[\nu_l^{(2)}(u_k)\right]_{k=1,\ldots,M}^{l=1,\ldots,d} \tag{C8}$$

to be the matrix that contains the discretized basis elements over the same regular grid, u^*. Define

$\Delta v = v_2 - v_1$ and $\Delta w = w_2 - w_1$ such that $\Delta u = \Delta v \times \Delta w$, then we obtain the estimate of the $d \times d$ Gram matrix, $\mathbf{G} = \left[\left\langle \nu_i^{(2)}, \nu_j^{(2)} \right\rangle_{\mathbb{F}^{(2)}}\right]_{i,j=1}^{d}$, via rectangular integration by computing

$$\mathbf{G} = \Delta u \mathbf{B}^\top \mathbf{B}. \tag{C9}$$

In addition, we obtain the $d \times N$ matrix of coefficients, $\mathbf{C} = \left[\left\langle y_t^{(2)}, \tilde{\nu}_l^{(2)} \right\rangle_{\mathbb{F}^{(2)}}\right]_{l=1,\ldots,d}^{t=1,\ldots,N}$, via least-squares with

$$\mathbf{C} = \mathbf{G}^{-1}\mathbf{B}^\top \mathbf{X}. \tag{C10}$$

The reader will notice that equation (C9) estimates d^2 integrals using rectangular integration. Similar to the one-dimensional case, we extend the trapezoidal rule to handle two-dimensional integration over a regular grid. Define the vectors $\mathbf{a}_1 = \left[1, \underbrace{2, \ldots, 2}_{M_1 - 2 \text{ times}}, 1\right]^\top \in \mathbb{R}^{M_1}$ and $\mathbf{a}_2 = \left[2, \underbrace{4, \ldots, 4}_{M_1 - 2 \text{ times}}, 2\right]^\top \in \mathbb{R}^{M_1}$ and define the diagonal $M \times M$ matrix, $\mathbf{D}$, whose main diagonal is the vector

$$\left[\mathbf{a}_1 \underbrace{\mathbf{a}_2 \cdots \mathbf{a}_2}_{M_2 - 2 \text{ times}}, \mathbf{a}_1\right]^\top \in \mathbb{R}^M.$$

With $\mathbf{D}$, we are able to define the following two-dimensional trapezoidal rule to estimate the Gram matrix as

$$\mathbf{G} = \frac{\Delta u}{4}\mathbf{B}^\top \mathbf{D}\mathbf{B}.$$

Rfssa fts Class

In this Subsection, we develop the S4 class, **fts**, where objects of this class have attributes **@coefs**, **@B**, and **@grid**. We consider only our bivariate FTS example developed thus far to illustrate objects of the class **fts** but the work can be generalized to account for p variables. The attribute, **@coefs**, will be a list of length 2 where the first entry of the list will contain the matrix seen in equation (C5) and the second entry contains the matrix seen in equation (C10). The attribute, **@B**, will be a list of length 2 where the first entry of the list will contain the matrix given by equation

(C3) and the second entry of the list will contain the matrix seen in equation (C8). The attribute, **@grid**, will be a list where the first element of the list will contain the grid given in equation (C2) and the second element of the list will contain the grid given in equation (C7).

In order to create an object of class **fts**, we invoke the constructor **fts()** which has arguments **X**, **B**, and **grid**. We have that **X** is a list of length 2 where the first element in the list is the matrix shown in equation (C1) and the second element in the list is the matrix given by equation (C6). We have that **B** is the same as **@B** and that **grid** is the same as **@grid**. Using objects of the class **fts**, we are able to employ our FSSA and MFSSA algorithms without having to rely on the **fda** package. We simply estimate the Gram matrix, **G**, for both variables using integration techniques discussed earlier and using these Gram matrices, we are able to estimate the projections onto the dual basis elements for each variable.